NR INDEKSU 353965
PL ISSN 0860-1674

8

Z MIKROKOMPUTEREM NA TY

Bajtek

MIESIĘCZNY DODATEK DO SZTANDARU MŁODYCH NR 8 (32) SIERPIEŃ 1988 CENA 150 ZŁ

PIANINO

ATARI 1040 ST

PROGRAMY SPOD NUTKI

CHIP NA PIĘCIOLINII

NUTKI

TIMEX I 64 KOLUMNY

KONKURS MATEMATYCZNY

MIEJSKA ESKAPADA



DECYDUJĄCE OGNIWO

Wyobraźcie sobie taką scenę: pustynia, jak okiem sięgnąć nie widać nic tylko piasek. W tym morzu piasku jedynym cywilizowanym elementem jest budka telefoniczna z maleńką satelitarną anteną na dachu. Zasilanie — oczywiście z baterii słonecznej. Do budki podjeżdża Beduin, wyjmuje z bagażnika swój przenośny komputer i podłącza go do telefonu. Po uzyskaniu potrzebnych informacji z położonego gdzieś na innym kontynencie banku danych, i po zapisaniu ich na dyskietkach, pakuje komputer i odjeżdża. W tym konkretnym przypadku Beduin przyjechał samochodem, ale równie dobrze mógł przyjechać wierzchem, na wielbłądzie...

Nie jest to żadna sytuacja wydumana, tylko prawdziwy obrazek, jaki w swych wędrówkach po świecie napotkał profesor Władysław Turski. Opowiedział o nim pani redaktor Bożenie Kastory, a ta zamieściła go w wywiadzie z Profesorem zamieszczonym w „Życiu Warszawy". Ponieważ „Bajtek" od chwili swego powstania wiernie kibicuje utarczkom Profesora o spożytkowanie w Polsce przewag, jakie innym już od dawna daje informatyka, więc odnotowanie publicznych wystąpień Profesora uważamy niejako za swój statutowy obowiązek.

Otóż główna teza profesora Turskiego (pominmy zawartą w wywiadzie celną, lecz niestety od lat niezmienną diagnozę stanu naszej informatyki) brzmi następująco: nic się nie zmieni, dopóki nie będziemy mieli w Polsce oficjalnego programu rozwoju informatyki. Ma taki program polska elektronika (nie jest to, dodajmy, żaden program rewelacyjny, ale jest i nie pozwala zdechnąć naszej elektronice do końca), nie ma takiego programu skoligacona z nią całkiem niedalekimi powiązaniami rodzinnymi informatyka.

My się strasznie kochamy w informatyce — ironizuje Profesor — tyle, że romantycznie. Nikt nie próbuje nawet dokonać wyboru, gdzie da się ją zastosować, a gdzie jeszcze przez długie lata będzie to niemożliwe. Czy będzie się stosować komputery w bankach czy w zarządzaniu, a może w służbie zdrowia, w edukacji czy w projektowaniu inżynierskim. Natomiast niech nikt mi nie mówi, że we wszystkim będziemy stosować informatykę harmonijnie — bo nas po prostu na to nie stać. Ani finansowo, ani organizacyjnie.

Rezultatem tego braku koordynacji działań, czy też, mówiąc inaczej, braku programu rozwoju informatyki, jest sytuacja, panująca na przykład w polskim przemyśle. Odwiedzając polskie fabryki wszędzie spotkać można dzisiaj jakieś komputery. Tylko, że zazwyczaj są to komputery samotne i bezradne, bo kupowano je korzystając z nadarzających się okazji z myślą o tym, że będą sterować procesami produkcyjnymi, zautomatyzowaną produkcją itp. Nikt z kupujących nie wziął pod uwagę, że aby zacząć mówić o zautomatyzowanych fabrykach, o robotyzacji z prawdziwego zdarzenia, należy najpierw mieć podstawowe urządzenia automatyki, nie mówiąc już o robotach. Inaczej komputer czyli jednostka centralna, mózg całego układu, pozbawiony jest mięśni i nie ma czym sterować! A tych urządzeń automatyki jak nie było tak nie ma.

Piszę o tym w „Bajtku" nie gwoli narzekania — bo jest to akurat cecha zdecydowanie obca mojej naturze. Wychodzę po prostu z założenia, że skoro zaniedbania w naszej informatyce mierzą się latami (może nawet kilkunastoma), to i ich odrabianie trochę potrwa. Robotę tę będą musieli więc przede wszystkim wykonać dzisiejsi Czytelnicy „Bajtka"! A skoro tak, to lepiej jak najwcześniej o tym wiedzieć i na serio się do tego przygotowywać. A to znaczy: nie tylko zdobywać wiedzę informatyczną i umiejętności posługiwania się komputerem, ale również, poprzez dyskusje w gronach rówieśniczych, poprzez spotkania z tymi, którzy już się w bojach o polską informatykę nieco zahartowali, tworzyć pokoleniowy program zmian, pokoleniowy potencjał modernizacji naszej gospodarki i, jeśli okaże się to niezbędne, pokoleniowy program modernizacji naszych instytucji społeczno-politycznych. Temu właśnie celowi podporządkowane są m.in. obozy „Awangarda XXI", wśród uczestników których Czytelnicy „Bajtka" stanowią przecież przeważającą większość.

Mam uzasadnioną nadzieję, że jest to większość, która nadaje ton obozowym dyskusjom. Bo szanując wszystkie dziedziny wiedzy, wszystkie rodzaje twórczej aktywności, trzeba zawsze starać się wyraźnie widzieć, gdzie znajduje się sedno sprawy, ogniwo decydujące na danym etapie rozwoju. Wszystkie rozwinięte kraje świata — w tym również te, które do tej grupy niedawno doszlusowały, — zgodnie już od dawna uznały, że ogniwem tym jest obecnie informatyka.

Waldemar Siwiński

Bajtek

„BAJTEK" — MIESIĘCZNY DODATEK DO „SZTANDARU MŁODYCH"
ADRES: 00-687 Warszawa, ul. Wspólna 61. Tel. 21-12-05 Przewodniczący Rady Redakcyjnej: Jerzy Domański — redaktor naczelny „Sztandaru Młodych".
ZESPÓŁ REDAKCYJNY: Waldemar Siwiński (z-ca redaktora naczelnego „SM" — kierownik zespołu „Bajtka"), Grzegorz Onichimowski (sekretarz redakcji „Bajtka"), Roman Poznański (kierownik działu klanów), Krzysztof Czernek, Sławomir Gajda (red. techniczny), Andrzej Pilaszek, Sławomir Polak, Wanda Roszkowska (opr. graficzne), Kazimierz Treger, Marcin Waligórski, Roman Wojciechowski. Zdjęcia w numerze: Leopold Dzikowski
klany redagują:
Commodora — Klaudiusz Dybowski, Dominik Falkowski
Amstrad-Schneider — Jonasz Mayer
Spectrum — Marcin Przasnyski
Atari — Wojciech Zientara, Sergiusz Piotrowski

Fotoskład — Tadeusz Olczak,
Montaż offsetowy — Grażyna Ostaszewska,
Korekta — Maria Krajewska, Zofia Wółtańska
WYDAWCA: RSW „Prasa-Książka-Ruch" Młodzieżowa Agencja Wydawnicza, al. Stanów Zjednoczonych 53, 04-028 Warszawa. Telefony: Centrala 13-20-40 do 49, Redakcja Reklamy 13-20-40 do 49 w. 403, 414.
Cena 150 zł.
Skład techniką CRT-200, przygotowalnia offsetowa i druk: PRASOWE ZAKŁADY GRAFICZNE RSW „PRASA-KSIĄŻKA-RUCH" w Ciechanowie, ul. Sienkiewicza 51.
Nr zlecenia 055528 n. 150.000 egz. U-113

ZA MIESIĄC

We wrześniowym „Bajtku" znajdziecie m.in.:

- **komputer w szkole, czyli recenzje polskich programów edukacyjnych, dyskusję o problemach komputeryzacji rodzimej oświaty, wieści ze szkół amerykańskich**
- **nowa edycja konkursu o „Złotą Dyskietkę Bajtka"**
- **nowe i stare wyroby ze znakiem Commodore**
- **Atari Mega St**
- **Pyramania**



FLETY DLA PASTUSZKÓW

rozmowa z Wojciechem Mannem, dziennikarzem muzycznym

— Czy w audycji *), którą Pan prowadzi równolegle z naszą rozmową będzie „kawałek" muzyki komputerowej?

— W pierwszej chwili chciałem powiedzieć — nie. Ale jeśli damy muzykę nagraną w tym roku na nowych płytach, to „kawałek" ten jest nieunikniony. Zawsze gdzieś tam komputer się odezwie. Na przykład na jednej z płyt, którą będę grał dzisiaj jest komputerowy perkusista. Wiele nagrywanych obecnie albumów to kombinacje twórczości żywego człowieka i maszyny. Nie mówiąc już o tym, że gramy płyty kompaktowe, gdzie komputer zamieszany jest z racji sposobu rejestracji.

— Ale tak naprawdę — Panie Wojtku — czy widział Pan grający komputer?

— Widziałem, owszem, urządzenia, które wydają jakieś tam zaprogramowane dźwięki. Może jestem zbyt konserwatywnie nastawiony, jeśli chodzi o całkowite oddawanie się technice — ale takiej, której słyszałem, graniem nie było.

— Uściślijmy zatem, co kryje się pod rozpowszechnionym już pojęciem „muzyka komputerowa"?

— Rzeczywiście, dalekie do precyzji w tym sformułowanie, które sugeruje, że jest to muzyka tworzona przez komputer. Odpowiednio zaprogramowany, nafaszerowany ogromną ilością danych dotyczących harmonii, następstw nut, itd. czy też jakiś olbrzym science fiction, który po przeanalizowaniu dzieł wszystkich wielkiego kompozytora wyciągnąłby z tego uniwersalny wniosek — rodzaj muzycznego algorytmu i następnie stworzył coś zupełnie nowego.

— Czyli precyzując...

— „Muzyką komputerową" potocznie określa się utwory, które powstały przy pomocy różnorodnych urządzeń elektronicznych przetwarzających dźwięk.

— Jeden z amerykańskich producentów stwierdził niedawno, że jeszcze pięć lat temu znał wszystkich, którzy uprawiali ten rodzaj twórczości. Dzisiaj natomiast — ubolewał — zna kilku, którzy jeszcze nie podłączyli się do mikroprocesora. Skąd ten pęd artystów ku elektronice? Koniunktura?!

— Nie jestem muzykiem; mogę tylko jako obserwator i świadek wymienić parę przyczyn, z których na pierwszym miejscu stawiałbym lenistwo i niedouczenie. Wielu ludzi, którzy nie potrafią dobrze grać na klasycznym instrumencie oszukuje się podpierając się tym, co im technika podsuwa.

— Są tez zdolni?

— I dodałbym — ambitni twórcy poszukujący nowych brzmień, innej formy muzycznej wypowiedzi. W udziwnianiu jak skrzypce, fortepian, gitara i probują to przetwarzać. Publiczność częściem dobrze przyjmuje takie próby — częściej jednak z rezerwą. Poza wszystkim zaś, jest to pęd człowieka ku nowemu, do eksperymentu, do uzyskania nowych jakości. Myślę, że jest to w porządku, bo z tego bałaganu coś tam w końcu się stworzy nowego, pięknego, cennego... Z polskich muzyków najwcześniej chyba Michał Urbaniak stosował najróżniejsze przystawki do swoich instrumentów.

— Wspomniał Pan o publiczności, ktora nie zawsze ma ochotę na eksperymenty. Moda na ten rodzaj twórczości nie daje jej wielkiego wyboru?

— Mimo to jestem przekonany, że to publiczność w końcu decyduje wynajdując swoim ósmym zmysłem to, co najwartościowsze. I na przykład — obecnie w Stanach duża grupa słuchaczy, przeważająca nawet, sięga po rozwiązania konserwatywne. Dlaczego tam tak świetnie sprzedaje się Sting, Bruce Springsteen, Bruce Hornsby — to są ludzie, którzy wcale nie szaleją na punkcie techniki.

— Przesyt?

— Tak — chociaż jest coś takiego jak New Age, tak określa się nagrania rejestrowane przy użyciu tych wszystkich nowomodnych wynalazków. Jest taka stacja w Los Angeles, która nazywa się The Wave i nadaje tę muzykę na okrągło. Ale okazuje się, że zjawisko to stało się już tym co określamy jako „muzak" — tło. Nie jest to muzyka, którą się ludzie fascynują; istnieje gdzieś tam w drugim planie uprzyjemnia jazdę samochodem, czekanie w kolejce.

— Ale przecież znane są utwory, które fascynują, nazwiska muzyków, które przyciągają!

— Ci, którzy pojęli na czym polega współpraca z, ogólnie mówiąc, komputerem, korzystają z niego jako z bardzo nowoczesnego warsztatu kompozytorskiego. Na początku i oni przeszli okres fascynacji możliwościami sprzętu, który daje nieodkryte dotąd możliwości. Po pewnym czasie jednak każdy prawdziwy kompozytor zaczyna traktować komputer jak jeszcze jeden instrument. Najważniejsze jednak jest to, co tworzy się przy jego pomocy. Jeśli elektronika wspomaga go w tym, co on sam myśli, to wszystko jest O.K. Słyszymy tutaj na przykład, z drugiego pokoju, muzykę Uriah Heep. To jest początek lat 70, kiedy nie było jeszcze tych wynalazków. Do czego zmierzam — otóż istnieje taka teoretyczna możliwość, że dobry programista kompozytor siądzie do maszyny i uzyskując takie same brzmienia zrobi to samo. Jeśli tak, to tu zaczynam się buntować. Jeżeli używa się czegoś, co imituje dźwięki możliwe do uzyskania drogą naturalną, klasyczną, to oznacza, że zaczynamy kręcić się w kółko.

— Warto tu chyba jeszcze wspomnieć o pewnej wygodzie, która pozwala artyście angażować do nagrania via komputer najlepsze smyczki, gitary, ba, całą nawet orkiestrę — do znudzenia, za darmo, nie licząc wartości sprzętu.

— W tym momencie wchodzimy w sferę produkcji komercyjnej, która może być doskonała, nagrana na kompakcie — ale wtedy ja proszę o możliwość wyboru. Jeżeli nie interesuje mnie piosenka stworzona na drutach i kablach, tylko chcę usłyszeć żywych ludzi, to muszę mieć taką możliwość.

— Dodałbym do tego jeszcze tę denerwującą niektórych hiperjakość nagrania elektronicznego, która wręcz odczłowiecza wrażenia.

— Mam kolegę muzyka, który twierdzi, ze kompaktu nigdy nie kupi, bo dla niego cały smak i radość słuchania płyty to są te lekkie zniekształcenia, trzaski, szumy.

— Różnie słucha się muzyki. Jednym wystarcza samo wrażenie, innym do pełnego odbioru konieczna jest wizualizacja muzyki, orkiestry. A jeśli wiadomo, że gra komputer?

— Wśród ludzi słuchających muzyki zauważyłem różne formy podejścia do jej odbioru. Są tacy, którzy chcieliby sobie wizualizować i uzmysławiać, że to gra mistrz którego się zna lub pozna. Inni koncentrują się na słuchaniu muzyki jako takiej, zakładając, ze jest dobrze wykonana, szukając w niej tego co znaleźć można w dobrej literaturze, książce bez ilustracji — własnych obrazów, nastrojów. Wtedy ich wyobraźnia jest pobudzona najbardziej. A są też wreszcie tacy, którzy w ogóle nie słuchają muzyki, niczego sobie nie wyobrażają, a tylko sprawdzają, czy nie trzeszczy, zgrzyta i czy jest przenoszone całe pasmo.

— Czy fachowiec od razu rozpozna tę muzykę?

— Można ją podrobić. Zdarzają się też sytuacje odwrotne: niedawno jeden z moich redakcyjnych kolegów słuchając nagrania rockowego, w którym była rozbudowana sekcja smyczków stwierdził: „idealnie zrobili to na semplerze!" Dopiero później dowiedział się, ze grali to prawdziwi — jak to śmiesznie brzmi — muzycy.

— Ostatnio Japończycy wymyślili rodzaj instrumentu elektronicznego z wyglądu przypominającego flet z czternastoma klawiszami. Jak twierdzą wynalazcy tego urządzenia, można na nim zagrać wszystko. Czy nie obawia się Pan, że przy takim tempie ewolucji, a raczej rewolucji w tej dziedzinie, za kilka lat straci Pan pracę na rzecz małej czarnej skrzynki z jednym tylko przyciskiem z napisem „MANN", ktora na życzenie słuchaczy zagra, zaśpiewa, powie coś od serca i do rozumu...

— Byłbym za tym, żeby te japońskie flety rozdać pastuszkom. A co do reszty — wie pan — to tak, jak ogląda się kosmonautów jedzących befsztyk z tuby, sałatę z tuby i zupę pomidorową z tuby. Niech oni sobie to jedzą. Ale na prawdziwy, porządny befsztyk zawsze znajdzie się amator, ktoś kto go zechce przyrządzić i ktoś, kto go zje ze smakiem. I będą prawdziwe flety i sztuczne flety, głęboko wierzę w to, że ludzkość utrzyma tu właściwą proporcję.

— I jest Pan przekonany, ze nie dojdziemy do momentu, kiedy zacznie się dzielić muzykę na tę dobrą, złą i komputerową?

— Mam nadzieję, gdyby tak się stało, to ja się zewsząd wypiszę.

rozmawiał: Franciszek Penczek



Po pewnym czasie prawdziwy kompozytor zaczyna traktować komputer jako jeszcze jeden instrument

* — rozmowa odbyła się w trakcie (przerwach) cosobotniej audycji radiowej Wojciecha Manna pt. „RadioMann"

PIANINO

Mimo nienajgorszych możliwosci dźwiękowych komputerów serii Atari, niewiele do tej pory pojawiło się w „Bajtku" (a także w innych rodzimych czasopismach) programów wykorzystujących ten fakt. Bywalcy giełd wiedzą, że spotkać tam można duże programy umożliwiające tworzenie własnych melodii korzystając z zapisu nutowego (np. SoundMachine), czy też generujące dźwięki przy poruszaniu joystickiem. Program, który chciałbym dziś zaprezentować, zamienia nasz komputer w klawiaturę syntezatora (co prawda o dość ubogich możliwościach). Klawiatura ma układ podobny jak w każdym instrumencie, z tym, że podzielona jest na dwie części.

Przyporządkowanie jest następujące: — klawisze najniższego i drugiego od dołu rzędu — to oktawy wielka i część małej (klawisz "Z" — C, "S" — Cis, "X" — D, itd.)— — klawisze rzędów najwyższego oraz drugiego od gory (poczynając od tabulatora) — oktawy mała (ponownie) i razkreślna ("Tab" — c, "1" — cis, "Q" — d,... i dalej: "U" — c', "I" — d', itd.)

Program wykorzystuje dwa rodzaje przerwań: przerwania klawiatury, należące do grupy maskowalnych (IRQ) oraz tzw. opóźnione przerwania synchronizacji pionowej (niemaskowalnej — NMI). Z tego powodu nie angażuje on procesora całkowicie, mozliwe jest więc wykonywanie równocześnie innego programu, chociażby napisanego w Basicu (np. dodatkowe efekty perkusyjne). Efektem ubocznym jest fakt, iż po wywołaniu bezpośrednim (USR(16909)) na ekranie pojawia się komunikat READY, który kontrastuje z powstałym rysunkiem klawiatury. Aby tego uniknąć zalecałbym użycie krótkiego programu:

```
1 X=USR(16909)
2 GOTO 2
```

Powrót do normalnej obsługi klawiatury możliwy jest po wciśnięciu kombinacji klawiszy: SHIFT+CONTROL+ESC.

Natężenie dźwięku jest przedstawione na ekranie w postaci słupka pojawiającego się w miejscu odpowiadającym wciśniętemu klawiszowi. Korzystanie z programu już po krótkim treningu nie powinno sprawiać żadnych kłopotów.

Andrzej Biazik

— LISTING 1 —

```
RH  10 A=16384:FOR I=0 TO 68
RJ  20 S=0:FOR J=0 TO 9:READ B:POKE A+J,B:
    S=S+B:NEXT J
VC  30 READ SK:IF S<>SK THEN 60
RE  40 A=A+10:NEXT I
HQ  50 ? " GOTOWE":END
LU  60 ? " BLAD W LINII ";100+10*I
IY  70 LIST 100+10*I:END
GW  100 DATA 65,8,128,0,0,0,0,0,158,0,359
RO  110 DATA 6,97,22,235,238,244,23,0,121,
    0,986
MN  120 DATA 0,163,177,131,235,0,0,4,0,219
    ,949
EC  130 DATA 128,65,193,0,214,209,0,69,60,
    0,956
SB  140 DATA 219,0,11,44,193,233,60,214,64
    ,0,1038
UX  150 DATA 0,0,130,161,116,109,0,162,255
    0,955
SP  160 DATA 0,215,130,0,26,30,22,0,0,0,42
    2
VP  170 DATA 0,0,10,0,10,13,7,11,8,7,66
AR  180 DATA 40,0,42,0,0,35,47,53,18,0,235
[illegible]B  190 DATA 0,15,0,18,22,25,26,0,23,31,15
    6
KC  200 DATA 0,26,21,0,17,0,20,14,26,15,14
    1
UX  210 DATA 21,23,11,0,0,0,7,12,9,8,91
TX  220 DATA 0,33 44,0,0,37,50,0,13,10,167
25  230 DATA 15 0,0,0,0,0,26,0,20,24,98
AB  240 DATA 35,26,31,33,5,0,4,0,0,7,141
YH  250 DATA [illegible],0,18,0,0,22,0,20,15,13,90
EU  260 DATA 12,0,14,9,0,11,18,0,19,0,81
YA  2[illegible]0 DATA 1[illegible],23,12,21,16,14,27,0,0,0,13
    [illegible]
HG  280 DATA 34 25,30,32,0,6,3,0,0,6,136
UT  290 DATA 1,0,136,72,152,72,174,9,210,2
    24,105,[illegible]
KE  300 DATA 220,208,40,169,0,141,14,212,1
    69,25,1207
FA  310 DATA 141,8,2,169,252,141,9,2,169,1
    38,1031
QL  320 DATA 141,36,2,169 194,141,37,2,169
    ,2[illegible]5,1146
JO  330 DATA 141,14,212,169,224,141,244,[illegible],
    3[illegible],32,1[illegible]1
PV  340 DATA 244,169,160,141,1,210,141,3,2
    10,56,1335
FX  350 DATA 1[illegible]8,30,1[illegible]8,0,64,[illegible],40,141,[illegible],
    21[illegible],1[illegible]9[illegible]
ZR  360 DATA 169,64,64,141,2,210,169,128,6
    4,133,1164
DG  370 DATA 20[illegible],169,60 141,6,210,169,160,
    141,1,1262
MQ  380 DATA 210,169,1[illegible],141,3,2,0,169,1,1
    41,241,1460
ZF  390 DATA 2,104,168,104,17[illegible],104,64,169,
    0,141,1026
HB  400 DATA 241,2,56,165,89,176,240 169,0
    ,133,1271
DF  410 DATA 203,133,205,169,224,133,204,1
    69,128,133,1701
GV  420 DATA 206,162,3,160,0,1[illegible],[illegible] 145,2
    05,200,1461
VK  430 DATA 208,249,230,204,230,206,[illegible] 1
    6,242,169,1956
OM  440 DATA 120,141,244,2,133,206,169,0,1
    33,205,1361
MB  450 DATA 160,6,169,127,145,205,200,192
    ,16,208,1430
QH  460 DATA 249,169,254,145,205,200,192 2
    4,208,249,1695
SU  470 DATA 169,231,145,205,200,192 [illegible],[illegible]
    8,240,169,1600
NH  480 DATA 0,145,205,73,255,200,192,40,2
    08,247,1565
ZX  490 DATA 32,32,244,24,165,88,10[illegible],128,1
    33,203,1154
TY  500 DATA 165,89,105,2,133,204,169,107,
    133,205,1402
QV  510 DATA 169,65,133,206,32,169,65,169,
    233,133,1374
FV  520 DATA 205,169,65,133,206,162,3,160,
    35,177,1316
FY  530 DATA 205,145,203,136,16,249,24,165
    ,203,105,1451
EB  540 DATA 40,133,203,165,204,105,0,133,
    204,202,1369
VO  550 DATA 16,231,96,1,0,128,0,2,1,0,475
AZ  560 DATA 128,0,128,0,2,1,0,128,0,2,389
ET  570 DATA 1,0,128,0,128,0,2,1,0,128,388
AV  580 DATA 0,2,1,0,128,0,128,0,2,1,262
XN  590 DATA 3,128,3,2,1,3,128,3,128,3,402
OZ  600 DATA 2,1,3,128,3,2,1,3,129,3,274
JN  610 DATA 128,3,2,1,3,128,3,2,1,3,274
GN  620 DATA 128,3,128,3,2,104,32,51,65,16
    9,665
VU  630 DATA 0,141,14,212,169,192,141,6,2,
    169,1046
MU  640 DATA 64,141,9,2,169,61,141,36,2,16
    9,794
PF  650 DATA 66,141,37,2,169,255,141,14,21
    2,169,1206
EE  660 DATA 0,141,198,2,169,160,141,1,210
    ,141,1163
28  670 DATA 3,210,96,173,241,2,240,4,169
    15,1153
[illegible]  680 DATA 133,204,24,165,204,240,[illegible],190
    204,105,1479
DU  690 DATA 160,141,[illegible],210 24,[illegible]65 [illegible]05,101
    88,105,1202
NF  700 DATA 4[illegible],133 2[illegible]7 165,89,[illegible],[illegible],133,[illegible]
    08,160,1240
9Z  710 DATA 0,16[illegible],14,169,[illegible],145,2[illegible],24,16[illegible]
    ,207,[illegible]093
YD  7[illegible] DATA [illegible]
    1[illegible],[illegible]08,1304
XW  730 DATA [illegible],16,[illegible]36,168 [illegible] 24[illegible] 24
    105 [illegible] 1449
XY  [illegible]40 DATA 1[illegible] 66,1[illegible] 86 1[illegible] 165,89,
    1[illegible]5,[illegible]
AN  750 DATA 133,208 [illegible]60,[illegible],169,4 1[illegible] 5
    6,1[illegible],124[illegible]
JJ  760 DATA [illegible]7,233 4[illegible] 133,[illegible] 165 [illegible]08,[illegible]3
    1,0,133,1559
[illegible]  770 DATA [illegible]08,[illegible] 16,238,165,2[illegible],133,20
    5,76,96,1[illegible]
1[illegible]  [illegible]80 DATA 220,[illegible],0 0,0,0,[illegible],0,0,0,220
```

— LISTING 2 —

```
0100        *= 16384
0110 KBCODE = 53769
0120 NMIEN = 54286
0130 VKEYBD = 520
0140 VVBLKD = 548
0150 CHBAS = 756
0160 CLEARS = 62496
0170 EXITVBLK = 58466
0180 AUDF1 = 53760
0190 AUDC1 = 53761
0200 AUDF2 = 53762
0210 AUDC2 = 53763
0220 AUDCNTL = 53768
0230 KEYDEL = 753
0240 P1   =   203
0250 P2   =   204
0260 P3   =   205
0270 P4   =   206
0280 EKRAN = 207
0290 PF2 =   710
0300 SCREEN = 88
0310 ;
0320 ;***********************************
0330 ;*     Instrument  klawiszowy     *
0340 ;*  dla Atari 800XL/65XE/130XE   *
0350 ;*                                *
0360 ;*    (c) 1988 Andrzej Biazik     *
0370 ;***********************************
0380 ;
0390 ; tablica nut : MB i SB
0400 ; F=896000/(MB*256+SB) (Hz)
0410 ;
0420 TAB1
0430 .BYTE 65,8,128,0,0,0,0,0
0440 .BYTE 158,0,6,97,22,235,238,244
0450 .BYTE 23,0,121,0,0,163,177,131
0460 .BYTE 235,0,0,4,0,219,128,65
0470 .BYTE 193,0,214,209,0,69,60,0
0480 .BYTE 219,0,11,44,193,233,60,214
0490 .BYTE 64,0,0,0,130,161,116,109
0500 .BYTE 0,162,255,0,0,215,130,0
0510 TAB2
0520 .BYTE 25,30,22,0,0,0,0,0
0530 .BYTE 10,0,10,13,7,11,8,7
0540 .BYTE 40,0,42,0,0,35,47,53
0550      .BYTE 18,0,0,15,0,18,22,25
0560      .BYTE 26,0,23,31,0,26,21,0
0570      .BYTE 17,0,20,14,26,15,21,23
0580      .BYTE 11,0,0,0,7,12,9,8
0590      .BYTE 0,33,44,0,0,37,50,0
0600 TAB3
0610      .BYTE 13,10,15,0,0,0,0,0
0620      .BYTE 28,0,29,24,35,28,31,33
0630      .BYTE 5,0,4,0,0,7,2,0
0640      .BYTE 18,0,0,22,0,20,15,13
0650      .BYTE 12,0,14,9,0,11,18,0
0660      .BYTE 19,0,17,23,12,21,16,14
0670      .BYTE 27,0,0,0,34,25,30,32
0680      .BYTE 0,6,3,0,0,6,1,0
0690 ;
0700 ; procedura obslugujaca
0710 ; przerwanie klawiatury
0720 ;
0730 KLAW
0740      TXA             ;odlozenie
0750      PHA             ;na stos
0760      TYA             ;wartosci
0770      PHA             ;indeksow
0780      LDX KBCODE
0790      CPX #220        ;sprawdzenie
0800      BNE ET1         ;czy wcisnieto
0810      LDA #0          ;SHIFT+CNTL+ESC
0820      STA NMIEN
0830      LDA #25
0840      STA VKEYBD
0850      LDA #252
0860      STA VKEYBD+1
0870      LDA #138        ;jesli tak,
0880      STA VVBLKD      ;to powrot
0890      LDA #194        ;do typowych
0900      STA VVBLKD+1
0910      LDA #255        ;wartosci
0920      STA NMIEN       ;zmiennych
0930      LDA #224        ;systemowych
0940      STA CHBAS
0950      JSR CLEARS
0960      LDA #160        ;wylaczenie
0970      STA AUDC1       ;dzwieku
0980      STA AUDC2
0990      SEC
1000      BCS ET2
1010 ET1  LDA TAB1,X      ;jesli nie,
1020      BEQ ET3         ;to wczytaj
1030      STA AUDF1       ;mlodszy bajt
1040      LDA TAB2,X      ;okresu dzwieku
1050      STA AUDF2
1060      LDA TAB3,X      ;gdy rowny 0,
1070      STA P1          ;to kasuj KEYDEL
1080      LDA #60
1090      STA AUDCNTL
1100      LDA #160        ;gdy rozny od 0,
1110      STA AUDC1       ;to ustaw KEYDEL
1120      LDA #175        ;i wpisz okres
1130      STA AUDC2       ;dzwieku do
1140      LDA #1          ;AUDF1 i AUDF2
1150      STA KEYDEL
1160 ET2  PLA             ;zakonczenie
1170      TAY             ;obslugi
1180      PLA             ;przerwania
1190      TAX             ;klawiatury
1200      PLA
1210      RTI
1220 ET3  LDA #0
1230      STA KEYDEL
1240      SEC
1250      LDA SCREEN+1
1260      BCS ET2
1270 ;
1280 ; podprogram przepisujacy
1290 ; i modyfikujacy generator
1300 ; znakow oraz tworzacy
1310 ; rysunek klawiatury
1320 ;
1330 RYSUNEK
1340      LDA #0          ;przepisywanie
1350      STA P1          ;generatora
1360      STA P3          ;znakow -
1370      LDA #224        ;nowy
1380      STA P2          ;adres  32768
1390      LDA #128
1400      STA P4
1410      LDX #3
1420      LDY #0
1430 L1   LDA (P1),Y
1440      STA (P3),Y
1450      INY
1460      BNE L1
1470      INC P2
1480      INC P4
1490      DEX
1500      BPL L1
1510      LDA #128
1520      STA CHBAS
1530      STA P4
1540      LDA #0
1550      STA P3
1560      LDY #8
1570      LDA #127        ;tworzenie
1580 L2   STA (P3),Y      ;znakow
1590      INY             ;pomocniczych
1600      CPY #16         ;do rysunku
1610      BNE L2          ;klawiatury
1620      LDA #254
1630 L3   STA (P3),Y
1640      INY
1650      CPY #24
1660      BNE L3
1670      LDA #231
1680 L4   STA (P3),Y
1690      INY
1700      CPY #32
1710      BNE L4
1720      LDA #0
1730 L5   STA (P3),Y
1740      EOR #255
1750      INY
1760      CPY #40
1770      BNE L5
1780      JSR CLEARS      ;czysc ekran
1790      CLC
1800      LDA SCREEN
1810      ADC #176
1820      STA P1
1830      LDA SCREEN+1
1840      ADC #2
1850      STA P2
1860      LDA # <TAB4     ;zapelnianie
1870      STA P3          ;ekranu
1880      LDA # >TAB4     ;obrazem
1890      STA P4          ;klawiatury
```

```
1900      JSR ZAP
1910      LDA # <TAB5
1920      STA P3
1930      LDA # >TAB[illegible]
1940      STA P4
1950 ZAP  LDX #3
[illegible]60 L6   LDY #[illegible]
1[illegible]0 L7   LDA [illegible]P[illegible],Y
[illegible]0      STA [illegible]P1[illegible],Y
1990      DEY
[illegible]      BPL [illegible]
[illegible]10      C[illegible]
[illegible]0      LDA P[illegible]
2030      ADC #[illegible]
[illegible]      [illegible]TA P1
[illegible]      LD[illegible]
[illegible]      A[illegible] #[illegible]
[illegible]      [illegible]TA P[illegible]
[illegible]      [illegible]X
[illegible]      BPL L[illegible]
[illegible]      RTS
[illegible] ;
[illegible] ; dane dla obrazu klawiatury
[illegible]20 ;
[illegible]40 TAB4
[illegible]50      .BYTE 1,0,128,0,2,1
[illegible]60      .BYTE 0,128,0,128,0,2
[illegible]70      .BYTE 1,0,128,0,2,1
[illegible]      .BYTE 0,128,0,128,0,[illegible]
[illegible]      .BYTE 1,0,128,0,2,1
[illegible]      .BYTE 0,128,[illegible],1[illegible]8,0,[illegible]
[illegible]10 TAB5
[illegible]20      .BYTE 1,[illegible],128,[illegible],2,1
[illegible]      .BYTE 3,128,3,128,[illegible]
[illegible]      .BYT[illegible] 1,[illegible],12[illegible],[illegible],[illegible],1
[illegible]      .BYTE 3,128,3,1[illegible]8,3,2
[illegible]      .BYTE 1,[illegible],12[illegible],[illegible],2,1
[illegible]70      .BYTE 3,128,3,128,[illegible],[illegible]
[illegible]0 ;
[illegible]0 ; procedura inicjujaca
2300 ; opoznione przerwanie VBLK
[illegible]10 ; i zmieniajaca wektor
[illegible]0 ; obslugi przerwania
[illegible]30 ; klawiatury
[illegible]40 ; wywolanie : X=USR(16909)
[illegible] ;
[illegible]0 START
[illegible]70      PLA
[illegible]80      JSR RY[illegible]NEK
[illegible]90      LDA #[illegible]        ;ustawienie
2400      STA NMIEN       ;wektorow
2410      LDA #1[illegible]
2420      [illegible]TA VKEYBD
[illegible]30      LDA #[illegible]4
[illegible]40      [illegible]TA [illegible]KEYB[illegible]+1
[illegible]50      LDA # <VBLANK
[illegible]0      [illegible]TA VVBLKD
[illegible]70      LDA # [illegible]VBLANK
2480      [illegible]TA VVBLKD+1
[illegible]90      LDA #[illegible]
[illegible]      STA NMIEN
[illegible]      LDA #[illegible]
[illegible]      [illegible]TA P[illegible]
[illegible]30      [illegible]DA #1[illegible]
2540      [illegible]TA AU[illegible]1
[illegible]55[illegible]      STA AU[illegible]
[illegible]80      RTS
[illegible]70 ;
[illegible] ; program obslugi pr[illegible]erwania
[illegible] ; sy[illegible]hronizacji pionowej VBL[illegible]
[illegible]00 ;
2610 VBLAN[illegible]
[illegible]      L[illegible]A KEYDEL   ;[illegible]izer[illegible]
[illegible]      BEQ W[illegible]        ;czy [illegible]
[illegible]      LDA #15         ;przerwanie
[illegible]0      [illegible]TA P2          ;klawia[illegible]ry
[illegible] W1   C[illegible]
[illegible]70      LDA P2          ;modyfika[illegible]
[illegible]80      BEQ W2          ;natezenia
[illegible]90      DE[illegible] P2          ;dzwieku
2700 W2   ADC #160
[illegible]10      STA AUDC2
[illegible]      CLC
[illegible]30      LDA P3
2740      ADC SCREEN
[illegible]50      ADC #40
2760      STA EKRAN
[illegible]770      LDA SCREEN+1
[illegible]80      ADC #[illegible]
2790      STA EKRAN+1
[illegible]00      LDY #0
[illegible]10      L[illegible]X #14
2820 W3   LDA #0
[illegible]30      STA (EKRAN),Y
[illegible]40      CLC             ;kasowanie
2850      LDA EKRAN       ;poprzednio
2860      ADC #40         ;narysowanego
[illegible]70      STA EKRAN       ;slupka
[illegible]80      LDA EKRAN+1     ;natezenia
[illegible]0      ADC #0          ;dzwieku
[illegible]00      STA EKRAN+1
[illegible]10      DEX
[illegible]      BPL W3
[illegible]      LDX P2
[illegible]40      BE[illegible] W5
[illegible]      CLC
[illegible]      LDA P1
[illegible]      ADC [illegible]CRE[illegible]
[illegible]      AD[illegible] #[illegible]
[illegible]      [illegible]TA EKRAN
[illegible]      LDA SCREEN+1
[illegible]      ADC #[illegible]
[illegible]      [illegible]TA E[illegible]AN+1
[illegible]      LDY #0
[illegible]      [illegible]A #[illegible]
[illegible]      [illegible]TA [illegible]RAN),Y
[illegible]                      ;rysowanie
[illegible]      [illegible] EKRAN       ;slupka na
[illegible]      [illegible] #[illegible]         ;klawiatura
[illegible]      [illegible] [illegible]AN
[illegible]      [illegible] [illegible]AN+1
[illegible]      [illegible] #
[illegible]      [illegible] [illegible]AN+1
[illegible]      [illegible]                 ;zakonczenie
[illegible]      [illegible]                 ;procedury
[illegible]      [illegible]                 ;[illegible]k do
[illegible]      [illegible]                 ;[illegible]VB[illegible]
[illegible]
[illegible]
```

Podstawowym założeniem konstruktorów Atari 1040 ST było opracowanie komputera bazującego na sprawdzonym już modelu 520 ST, lecz o znacznie lepszych parametrach technicznych i użytkowych. Wprowadzono kilka istotnych zmian, które spowodowały poprawę walorów sprzętu.

Najważniejszą cechą nowego komputera jest prawie dwukrotnie większa pamięć. Zastosowanie 1024 KB RAM znacznie przyspiesza działanie licznych programów użytkowych, które mogą w ten sposób korzystać z bezpośredniego dostępu do danych bez częstego zapisywania i odczytywania ich z dysku. Oszczędność jest szczególnie widoczna w przypadku dużych baz danych. Sortowanie rekordów umieszczonych w RAM może skrócić się nawet o kilka minut. Oprócz użytkowników traktujących komputer jako narzędzie pracy zyskali także amatorzy zabawy. Udoskonalenie techniki animacji uzyskiwane dzięki buforowaniu kilku obrazów w pamięci oraz znacznie lepsze możliwości syntezy dźwięku wpłyną na atrakcyjność gier.

Drugą poważną zmianą jest wbudowanie napędu dysków elastycznych i umieszczenie zasilaczy pod obudową komputera. Wyeliminowano w ten sposób plątaninę kabli powstającą po zmontowaniu wszystkich urządzeń wchodzących w skład zestawu 520 ST, co jest nie bez znaczenia dla wygody obsługi. Dwie głowice stacji dysków o podwójnej gęstości pozwalają na zaformatowanie 3,5-calowej dyskietki do 720 KB. System operacyjny pozwala na zorganizowanie zapisywanych zbiorów w 40 katalogach i podkatalogach zagnieżdżanych na 8 poziomach. Bardzo mało prawdopodobne przekroczenie tego limitu może skończyć się skasowaniem niektórych danych lub nawet całego dysku. TOS nie sygnalizuje bowiem faktu założenia zbyt wielu katalogów. Oprócz podstawowego, wbudowanego napędu można dołączyć dodatkową stację dysków elastycznych lub dysk twardy. Najpopularniejsze są dysponujące pojemnością 20 MB twarde dyski SH204 i SupraDrive, lecz produkowane są również wersje o pojemności do 60 MB.

Kupując Atari 1040 ST musimy zdecydować się, który z dwóch oferowanych monitorów będzie lepiej spełniał nasze wymagania. Dwie niższe rozdzielczości, 320 × 200 punktów w 16 kolorach wybieranych z palety 512 i 640 × 200 punktów w 4 barwach wymagają zastosowania monitoru barwnego SC 1224. Najwyższa rozdzielczość 640 × 400 punktów jest osiągalna tylko na monitorze monochromatycznym (SM 124).

Klawiatura obsługiwana jest przez oddzielny procesor, zawiera ona wydzielony blok klawiszy numerycznych i sterujących kursorem oraz decydujące o charakterystycznym wyglądzie komputera klawisze funkcyjne.

Sterowanie przebiegiem programu jest możliwe nie tylko z klawiatury. Bardzo polecanym urządzeniem jest mysz. Dzięki niej użytkownik efektywnie wykorzystuje możliwości GEM, okna, technikę menu rozwijanych, ikony. Upraszcza to posługiwanie się programami graficznymi, edytorami tekstu, kalkulacyjnymi i wieloma innymi.

(j.j)



ATARI 1040 ST

DANE TECHNICZNE ATARI 1040 ST

PAMIĘĆ
1024 KB RAM
192 KB ROM

STRUKTURA WEWNĘTRZNA
Mikroprocesor 16/32 bitowy Motorola 68000, częstotliwość 8 MHz
Osiem 32 bitowych rejestrów danych
Dziewięć 32 bitowych rejestrów adresowych
16 bitowa szyna danych
24 bitowa szyna adresowa
Siedem poziomów przerwań
56 rozkazów, 14 trybów adresowania, 5 typów danych

PRZECHOWYWANIE DANYCH
Złącze do przyłączenia twardego dysku
Bezpośredni dostęp do pamięci 1.33 MB/s
Wbudowane gniazdo cartridge'a
Wbudowana 3,5 calowa dwustronna stacja dysków elastycznych o pojemności 720 KB

ZŁĄCZA
Złącze równoległe drukarki
RS 232 (V 24)
Złącze DMA dla drukarki laserowej i dysku twardego
Gniazdo do przyłączenia dodatkowego dysku elastycznego
2 gniazda joysticków
Wyjście video dla monitora RGB lub monitora monochromatycznego
Interfejs MIDI sprzęgający z urządzeniami syntetyzującymi dźwięk

KLAWIATURA
Klawiatura typu **qwerty**
Blok 18 klawiszy numerycznych
Klawisze sterujące ruchem kursora
Oddzielny procesor obsługujący klawiaturę
Dostępne wersje klawiatury angielskiej i niemieckiej

DŹWIĘK
3 generatory dźwięku
Częstotliwość generowanego dźwięku od 30 Hz do 16 kHz
3 kanały
Częstotliwość i głośność dźwięku każdego kanału odrębnie regulowana
Dynamiczna kontrola obwiedni dźwięku (ADSR)

GRAFIKA
Pamięć obrazu 32 KB
3 tryby rozdzielczości
320 × 200 punktów w 16 kolorach (najniższa rozdzielczość)
640 × 200 punktów w 4 kolorach (średnia rozdzielczość)
640 × 400 punktów monochromatycznych (najwyższa rozdzielczość)
512 kolorów

SYSTEM OPERACYJNY TOS z GEM
Możliwość jednoczesnego otwarcia do 4 okien
Menu rozwijane
Biblioteka procedur użytkowych GEM
Zegar czasu rzeczywistego

TARGI MUZYCZNE WE FRANKFURCIE

Jak co roku odbyły się we Frankfurcie nad Menem targi muzyczne, na których 890 wystawców z 38 państw pokazywało wszystko, co wiąże się pośrednio i bezpośrednio z dziedziną sztuki zwaną muzyką. Nie zabrakło na wystawie komputera ATARI serii ST — maszyny, która może symulować studio muzyczne bez konieczności korzystania z dodatkowych instrumentów, bez urządzeń muzycznych.

Na tegorocznych targach nie wystawiono niczego, co byłoby wcześniej nieznane z opisów w fachowej prasie. Pojawiło się natomiast bardzo dużo nowego oprogramowania przeznaczonego do tworzenia muzyki. Dotyczy to zwłaszcza ATARI ST.

Firma C-Lab zademonstrowała program pod nazwą „Notator", który jest kombinacją istniejącego już sekwencera „Creator" i programu drukującego nuty. Cechą szczególną tego softwaru jest nadzwyczajna szybkość. Wprowadzone z tzw. notesu nuty natychmiast pojawiają się na ekranie. To samo dzieje się z tekstem i specjalnymi znakami muzycznymi. Rzeczy niepotrzebne lub źle brzmiące usuwa się przy pomocy myszy. „Notator" może być używany tylko do edycji lub też może pokazywać na ekranie monitora równocześnie i nuty i edytor. Zależy to wyłącznie od woli użytkownika. Przy pomocy tego programu można kompleksowo opracować dowolną muzykę, dobrać właściwe otoczenie dla wybranych już nut, a na koniec wydrukować efekt własnej pracy. Firma C-Lab pokazała również rozszerzenie Midi-Output dla opisanego sekwencera. Program oparty jest na nowej metodzie edycji, która umożliwia korzystanie z czterech ścieżek równocześnie.

Konkurencją dla w/w programów może być nowa wersja Twenty Four Sequencer bardzo znanej firmy Steinbers. Ma ona możliwości niezwykłe. Dotychczasowy edytor nut i generator dźwięków zostały wzbogacone o edytor instrumentów perkusyjnych korzystający z trzech zestawów mogących zdefiniować 96 instrumentów. Program może czytać i zapisywać każdy utwór muzyczny w formacie zbioru Midi. Służy to wymianie różnych sekwencji. Firma pokazała również „Switcher", który pozwala na równoczesne trzymanie w pamięci komputera do dziesięciu różnych programów muzycznych i dowolne korzystanie z nich.

Nieliczne nowinki przedstawiła firma Hybrid Arts. Była to znana już, lecz poprawiona wersja programu EZ-Score Plus oraz Hybri-Switch" potrafiący trzymać w pamięci do 16 programów i szybko je przełączać w razie potrzeby.

Po raz pierwszy wystawiła się amerykańska firma SONUS. Pokazała ona bardzo dobry program do druku nut pod nazwą Super Score. Składa się on z 32 ścieżkowego sekwencera umożliwiającego aranżację 24 różnych sekwencji. Dochodzi do tego opcja edycji i transponowania poszczególnych dźwięków. Software umożliwia stosowanie specjalnych znaków muzycznych i tekstu, który można wstawić w dowolne miejsce nut. Wadą programu jest fakt, że może on korzystać tylko z drukarki 9-igłowej. SONUS pokazał także sekwencer Masterpiece i SST. Interesujące jest to, że we wszystkich produktach firmy daje się wykorzystać możliwość dołączenia drugiego równoległego wyjścia Midi.

Firma Dr T's pokazała nową wersję Keyboard Controlled Sequencer (KSC 1,6). Przedstawiono także software przeznaczony do komponowania muzyki (m.in. Programmable Variations Generator, który tworzy muzykę z podanego materiału melodycznego, posiada opcję pozwalającą dokładnie określić parametry zmian). Także program drukujący nuty „The Copyist" pojawił się w rozszerzonej wersji. Może współpracować z drukarką laserową i podawać informacje najważniejszym programom Desktop Publishing takim jak PageMaker, Ventura-Publisher i Publishing Partner Pro.

Francuska firma Di wystawiła software do tworzenia kompozycji na orkiestrę pod nazwą Big Band. Może on dowolnie generować rytm i poszczególne dźwięki wykorzystując różne kierunki stylów i tendencji muzycznych. W programie zawarty jest także prosty 24-ścieżkowy sekwencer.

Na koniec ciekawostka dla wszystkich zainteresowanych tematem Midi. **W Berlinie powstał klub Midi, wydający własną gazetę pod nazwą „Bits Music". Koszt uczestnictwa 45 DM rocznie.**

Adres klubu:
Midi e. V. Postfach 440308, 1000 Berlin 44.

Czytelników, którzy chcą dowiedzieć się więcej o wspomnianych w tekście programach muzycznych dla ATARI ST odsyłam do wskazanych firm. Zaznaczam jednak, że koszty opisanego oprogramowania kształtują się w granicach 1000 DM, a potrafią dochodzić do kilku tysięcy. Oferta ta jednak może zainteresować osoby profesjonalnie zajmujące się tą dziedziną sztuki.

Na podstawie ST Computer

Sergiusz Piotrowski

KROPKA NAD i

czyli jeszcze o zabezpieczaniu programów

O zabezpieczeniu programów na Atari pisano już na łamach „Bajtka" (wyd. specjalne, artykuł Piotra Adamczewskiego) i „Komputera" (nr 11/87, artykuł Piotra T. Grabczyńskiego). W żadnym jednak z artykułów nie postawiono przysłowiowej kropki nad „i". Dziś chciałbym podać kilka bardzo konkretnych porad i wskazówek dotyczących tego tematu.

Po co zabezpieczamy programy? Na swój użytek jest to i uciążliwe i niepotrzebne. Uzasadnionym natomiast powodem tego zabiegu jest publikacja programu na kasecie lub dyskietce. I tu właściwie nie można uchronić się przed „piratami-kopiarzami", jednak można skutecznie uniemożliwić złodziejski proceder plagiatorom. Przypomnę na początek ogólne zasady zabezpieczania programu napisanego w Atari Basic:

1) wyłączyć działanie klawisza BREAK,
2) przeprogramować RESET,
3) zabezpieczyć program przed błędami (TRAP),
4) uniemożliwić użytkownikowi listowanie programu.

Jeśli chodzi o klawisz BREAK, to najlepszym wyjściem jest umieszczenie POKE 566, 158 na początku programu, a POKE 566, 146 na końcu. Klawisz RESET najłatwiej przeprogramujemy przez POKE 580,1 na początku i POKE 580,0 na końcu. O klawiszu RESET pisano na tych łamach już sporo i w zasadzie można wybrać inną metodę. Ja jednak hołduję „zimnemu startowi" (może trochę ze złośliwości). Granice jednak złośliwości powinny nakazać, aby n i e z a p o m n i e ć o wpisaniu do komórki 580 zera na końcu programu. W przeciwnym wypadku możemy sprawić użytkownikowi niezbyt miłą niespodziankę! Nie będę też pisać o ochronie przed błędami, bo jest to w zasadzie zależne bardzo od konkretnego programu. Pamiętajmy jednak, że instrukcja TRAP w Atari Basic istnieje!

I wreszcie czwarty problem, a zarazem meritum tego artykułu — zabezpieczenie przed listowaniem. Sprawa listingu jest właściwie najważniejsza, bowiem ani przeprogramowanie klawisza BREAK, ani RESET nie pomoże, jeśli wydamy zlecenie LIST przed uruchomieniem programu. „Mapping the Atari" podaje (przedrukowany już kilkakrotnie w różnych miejscach) podprogram, którego wynikiem jest skutecznie i nieodwracalnie zabezpieczona przed listowaniem wersja zabezpieczanego programu. Podprogram dopisuje się na koniec (koniecznie) zabezpieczanego programu i wywołuje się przez GOTO. Na nośnik zostaje zapisany wówczas zabezpieczony już program. Wadą tej metody (zwłaszcza dla posiadaczy magnetofonów) jest to, że jedynym sposobem uruchomienia programu jest RUN „C:". Któż by wczytywał program kilkakrotnie dłużej, niż to jest potrzebne? I tu chcę postawić tę kropkę nad „i"...

Oto nowa wersja znanego już podprogramu (zamiast SAVE „C " jest CSAVE):

```
NJ   31000 FOR VARI=PEEK(130)+PEEK(131)*256
      TO PEEK(132)+PEEK(133)*256:POKE VARI,
     155:NEXT VARI
WF   31100 POKE PEEK(138)+PEEK(139)*256+2,0
     :CSAVE :NEW
```

1) linie 31 000 i 31 100 to ostatnie linie programu,

2) chcąc nagrać zabezpieczoną wersję piszemy GOTO 31 000,

3) w treści programu nie powinno być instrukcji LIST (efekt nieciekawy, możecie sprawdzić),

4) program powinien kończyć się NEW.

Tak zabezpieczony program nie będzie reagował ani na RUN, ani na LIST, ani na RUN „C:". Będzie jednak wspaniale działać, jeśli napiszemy dla niego specjalny program ładujący.

A oto ogólny schemat takiego programu

```
GM   0 POKE 566,158:POKE 580.1:REM przeprog
     ramowanie BREAK i RESET
CE   1 TRAP 100:REM zabezpieczenie przed bl
     edami ladowania
NT   2 DIM X$(40):POKE 764,33
PN   3 GOSUB 50:REM podprogram rysujacy str
     one tytulowa
SB   4 X$="[illegible]"
WA   5 X=USR(ADR(X$))
PH   50 ? "K":? "Loading...":RETURN
SG   100 GRAPHICS 0
AS   101 ? "Error in loading: ";PEEK(195)
YR   102 POKE 580,0:POKE 566,146:NEW
```

W programie ładującym należy wyrzucić komentarze i ewentualnie (w zależności od potrzeb) rozbudować podprogram strony tytułowej. W samym zabezpieczanym programie można na początku nie umieszczać jeszcze POKE 580,1 i POKE 566, 158 (choć lepiej umieścić — na wszelki wypadek), a także należałoby wyrzucić wszelkie komentarze (są zbędne, bo ewentualne poprawki można będzie dokonywać już tylko na niezabezpieczonej wersji — trzeba sobie taką zostawić!). Pamiętać należy o zakończeniu programu sekwencją POKE 580, 0:POKE 566, 146:NEW.

Andrzej-Piotr Szyndrowski

DESZYFRATOR DO SOUND MACHINE

Przedstawiony program umożliwia udźwiękowienie własnych programów napisanych w Basicu.

Przy pomocy programu SOUND-MACHINE piszemy muzykę, którą później będziemy chcieli wykorzystać we własnym programie. Zapisujemy ją na taśmie funkcją CREATE. Następnie uruchamiamy zamieszczony obok program i wczytujemy zapisaną muzykę. Po odczycie melodii komputer automatycznie zaczyna grać.

Po odpowiedniej modyfikacji zamieszczonego programu można dołączyć go do własnego programu. Muzyka rozbrzmiewa niezależnie od wykonywania programu w Basicu. Na koniec jeszcze jedna ważna informacja: program nie działa ze stacją dysków. Konflikt wynika z używania przez program tego samego obszaru pamięci co DOS.

Opis programu:

Linia 60
Wczytanie 15 bajtów pilotujących. We własnym programie w Basicu są one zbędne.

Linie 70-80
Wczytanie właściwego programu i melodii

Linie 100-120
Obsługa ewentualnych błędów

Linia 130
Wpisanie rozkazu PLA, który jest niezbędny dla prawidłowego działania instrukcji USR

Linia 140
Uruchomienie wczytanego programu oraz włączenie melodii. Zatrzymanie melodii uzyskujemy przez POKE 79661, a całkowite wyciszenie przez FOR I=O TO 3:SOUND I,O,O,O:NEXT I.

Jacek Górski

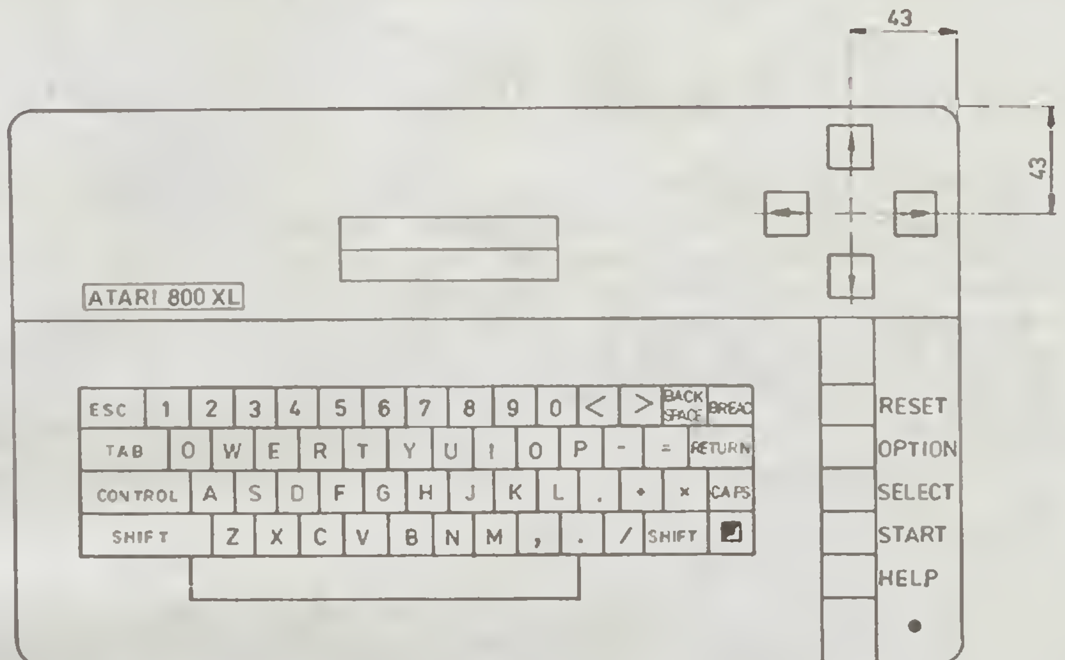
```
ZK   10 REM *     BITels     *
MR   20 REM * ACC SZCZECIN *
RE   30 POKE 82,0:POKE 752,1:? CHR$(125)
QZ   40 ? :? :? "                LADOWANIE
WH   50 OPEN #1,4,0,"C:"
LU   60 FOR I=1 TO 15:GET #1,Z:NEXT I
FG   70 TRAP 100:A=7944
SD   80 A=A+1:GET #1,X:POKE A,X:GOTO 80
YJ   100 IF PEEK(195)=136 THEN 130
PR   110 ? :? :? CHR$(125)
IX   120 ? "    BLAD NR ";PEEK(195):END
FA   130 POKE 7944,104:REM PLA
CG   140 P=USR(7944):POKE 7966,0
```

KLAWISZE KURSORA

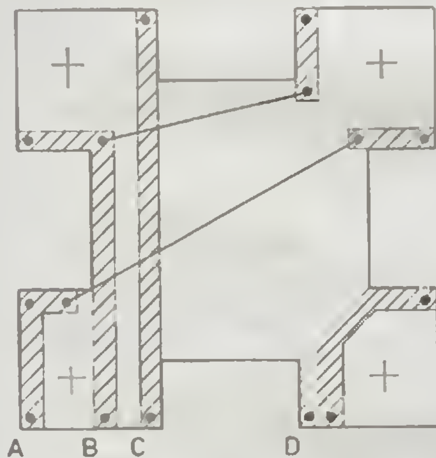
Chciałbym zaproponować wszystkim użytkownikom komputerów Atari 800 XL drobną przeróbkę umożliwiającą swobodniejsze operowanie kursorem na ekranie monitora.

W chwili obecnej dla przesunięcia kursora bez wymazywania tekstu konieczne jest posłużenie się jednocześnie dwoma klawiszami — CTRL oraz do wyboru jednym z klawiszy z rysunkiem strzałki określającej kierunek ruchu. W związku z tym konieczne jest używanie obu rąk jednocześnie. Moja propozycja dotyczy zamontowania 4 dodatkowych klawiszy umożliwiających zrealizowanie tej funkcji przy pomocy jednej tylko ręki.

Zamieszczone szkice wyjaśniają miejsce zamontowania dodatkowych klawiszy oraz widok płytki drukowanej, do której klawisze te są montowane. Nie podawałem na tych szkicach wymiarów, ponieważ zależą one od kształtu i wymiarów posiadanych klawiszy. Przy przeróbce swojego komputera posłużyłem się klawiszami z uszkodzonego sumatora. Kolejny rysunek to widok płytki drukowanej klawiatury komputera (widok od strony druku) oraz widok, czy raczej przekrój, wspornika do montażu płytki drukowanej z klawiszami.



Widok komputera wraz z dodatkowymi klawiszami.



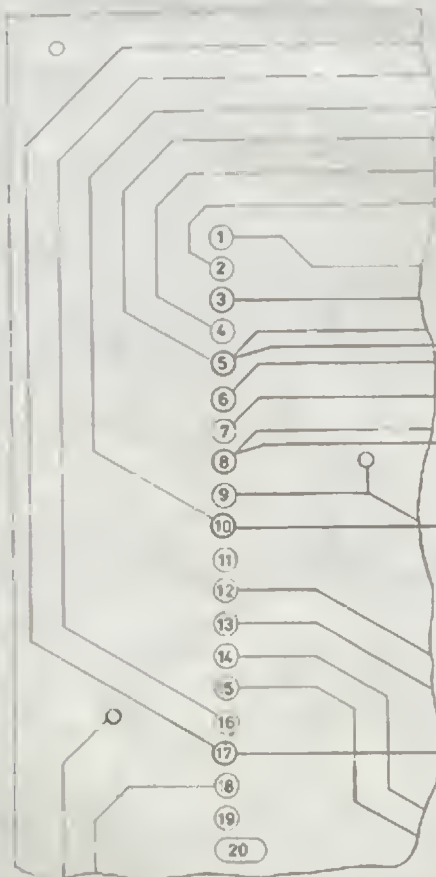
Płytka drukowana dodatkowych klawiszy (rysunek bez skali) — widok od strony druku. Dwie ciągłe linie — zwory: A, B, C i D — wyjścia.

Kolejność czynności przy przeróbce komputera jest następująca:

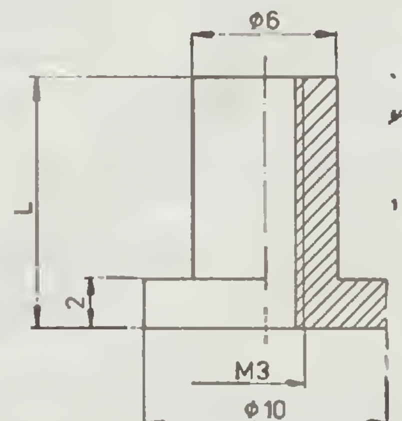
1. Wykonanie płytki drukowanej i zamontowanie na niej klawiszy.
2. Wykonanie otworów w pokrywie górnej komputera po uprzednim dokładnym pomiarze rozstawu klawiszy na płytce drukowanej.
3. Toczenie i montaż wsporników płytki drukowanej klawiszy — żywica „Distal" lub „Epidian".
4. Lutowanie przewodów zgodnie ze schematem

Tak przerobiony komputer użytkuję już od kilku miesięcy i bardzo chwalę sobie wprowadzone przeróbki. Przyznać również muszę, że nie jest mi znany autor tej zmiany, podpatrzyłem to rozwiązanie na giełdzie komputerowej we Wrocławiu, a przerobiony tak komputer został przywieziony z RFN.

Bogusław Kornafel



Schemat połączeń elektrycznych płytki drukowanej klawiatury komputera (widok od strony druku) i płytki dodatkowych klawiszy. Połączenia: A-8, B-10, C-5, D-17.



Wspornik do montażu płytki drukowanej z klawiszami (wymiar L należy tak dobrać, aby klawisze wystawały około 6 mm ponad obudowę komputera).

OD ŚRODKA

cz. VII

Cześć po raz ostatni!
W dzisiejszym odcinku mamy zamiar zająć się odkodowywaniem zakodowanych bloków typu "Bytes" (brzmi to trochę jak masło maślane, ale cóż robić?), oraz niepublikowanymi instrukcjami mikroprocesora Z-80. Na zakończenie — garść krótkich i prostych, lecz często używanych zabezpieczeń.

Na wstępie jednak niewielkie uzupełnienie do poprzedniego odcinka. Opisaliśmy w nim sposób uruchamiania się bloków kodu maszynowego przez wczytanie w obszar stosu maszynowego. Aby taki blok się uruchomił, wystarczy jedynie wpisać instrukcję LOAD ""CODE, która go załaduje. Aby przekonać się o tym w praktyce, wpisz i uruchom program z listingu 1

Nagra on na taśmę krótki blok kodu maszynowego, który będzie się sam uruchamiał. Po nagraniu tego bloku wykasuj pamięć przez **RANDOMIZE USR O** lub RESET (ewentualnie przestaw RAMTOP na normalną wartość przez **CLEAR 65367)** i wczytaj go wpisując LOAD ""CODE. Jego zadaniem jest poinformowanie o tym, że się uruchomił — robi to wysyłając kilka dźwięków. Usłyszysz je natychmiast po wczytaniu programu, gdy tylko z ramki znikną granatowo-żółte paski. To tyle tytułem uzupełnienia

Co to znaczy, że program lub jego blok jest zakodowany? Zakodowanie jest to rodzaj bardzo prostego szyfru uniemożliwiającego poprawne działanie programu. Do jego uruchomienia służy specjalna procedura odkodowująca, która jest umieszczona w tym programie oraz — co najważniejsze i chyba oczywiste — nie jest zakodowana. Czemu więc służy kodowanie? — Jest to po prostu utrudnianie dostępu do tekstu programu wtedy, gdy złamane zostały wcześniejsze zabezpieczenia i program jest wczytany bez uruchomienia, bowiem to, co znajduje się w pamięci nie jest programem, lecz jego „półproduktem", który dopiero po przetworzeniu przez procedurę odkodowującą staje się właściwym programem. Może to polegać np. na zanegowaniu wszystkich bajtów programu, a więc — jeśli w grze będziemy szukać disasemblerem kodu instrukcji DEC A lub podobnej, aby znaleźć „nieśmiertelność", to nasze poszukiwania spełzną na niczym, bo instrukcja ta zamiast kodu 3Dh będzie reprezentowana przez kod C2h. Dobrym przykładem jest znany wam zapewne program „Art Studio". Jego część BASIC-owa nie jest praktycznie w żaden sposób zabezpieczona, lecz główny blok programu ("studio-mc" CODE 26000, 30672) jest właśnie częściowo zakodowany. Co zrobić, aby go odkodować? — po prostu znaleźć adres, od jakiego blok ten jest uruchamiany. W przypadku „Art Studia" jest to adres 26000. Tam znajduje się instrukcja JP 26024, która skacze do procedury odkodowującej. Oto jej tekst:

```
26024 21C165  LD HL,26049   ;Zapisanie adresu
26027 E5      PUSH HL       :26049 na stos.
26028 21C165  LD HL,26049   ;Adres początku.
26031 11476C  LD DE,27719   ;Adres końca.
26034 7E      LD A,(HL)     ;Pobranie bajtu z
26035 D622    SUB 34        ;pamięci, przeko-
26037 07      RLCA          ;dowanie  go  po-
26038 EECC    XOR #CC       ;przez SUB, RLCA,
26040 77      LD (HL),A     ;XOR i odesłanie.
26041 23      INC HL        ;Następny adres.
26042 B7      OR A          ;Sprawdzenie  wa-
26043 ED52    SBC HL,DE     ;runku końca,skok
26045 19      ADD HL,DE     ;do pętli lub po-
26046 20F2    JR NZ.26034   ;wrót  pod  adres
26048 C9      RET           ;zapisany na sto-
26049 B2EDF1...             ;sie,czyli 26049.
```

Najpierw procedura umieszcza na stosie adres 26049. Wrócimy zaraz do tego. Teraz zaczyna się właściwe odkodowywanie: do rejestru HL ponownie ładowany jest adres 26049 — jako początek zakodowanego bloku, do DE — 27719 jako adres ostatniego zakodowanego bajtu. Następnie w pętli odkodowywane są kolejne bajty — instrukcje **SUB 34, RLCA** i **XOR #CC** są kluczem, według którego odszyfrowywana jest ta część programu. Wreszcie — sprawdzany jest warunek osiągnięcia adresu 27719 (zawartego w DE), który kończy odkodowywanie. Wykonywana jest instrukcja RET, ale ostatnim zapisanym na stosie adresem nie jest adres powrotu do **BASIC-a**, lecz zapisany na początku przez **PUSH HL** adres 26049, czyli adres dopiero co odkodowanego bloku, następuje więc jego uruchomienie.

Zwróćcie uwagę na to, jakie instrukcje dokonują deszyfracji, żadna z nich nie gubi ani jednego bitu. Odejmowanie przeprowadzane jest modulo 256 i dla dwóch różnych danych wejściowych wyniki też muszą być różne, **RLCA** zamienia wartości bitów 7, 6, 3, i 2. na przeciwne. Innymi instrukcjami mającymi tę własność są np. **ADD, INC, DEC, RRCA, NEG, CPL** ale nie **OR** ani **AND**. W jaki sposób więc należy odkodować ten blok? Najprościej — wpisując w **BASIC-u**:

POKE 26027,0 (kod instrukcji NOP)

likwidując tym samym instrukcję **PUSH HL** (RET na końcu procedury powróci do BASICA) i wykonać **RANDOMIZE USR 26000.** Warto jednak pamiętać o tym, że procedura odkodowująca może być sprawdzana przez inny fragment programu. W „Art Studiu" tak istotnie jest. Oto dalsza część programu.

```
26289 67      LD H,A        ;W A znajduje się
26284 6F      LD L,A        ;już wartość 0.
26285 E5      PUSH HL       ;Zapisanie jej na
26292 3AAB65  LD A,(26024)  ;stos i sprawdze-
26295 FE21    CP #21        ;nie   zawartości
26297 C0      RETNZ         ;komórek  o adre-
26298 2AA965  LD HL,(26025);sach 26024-26027
26301 B7      OR A          ;i  jeśli  są  one
26302 11C165  LD DE,26049   ;inne, niż powin-
26305 ED52    SBC HL,DE     ;ny być, to wyka-
26307 C0      RET NZ        ;sowanie  pamięci
26308 3AAB65  LD A,(26027)  ;poprzez  RET  NZ
26311 FEE5    CP #E5        ;(pod adres 0).
26313 C0      RET NZ     ;gdy wszystko jest OK
26314 E1      POP HL     ;to zdjęcie  adresu 0
26315 C9      RET        ;i „normalny" powrót.
```

Fragment ten sprawdza, czy procedura odkodowująca na pewno uruchomiła cały program, a jeśli nie — przez **RET NZ** kasuje pamięć (bo na stosie zapisany jest adres 0).

Cóż więc robić? Można kolejno likwidować te zabezpieczenia, ale w niektórych programach jest to bezcelowe. Wtedy pozostaje inne wyjście: ponownie zakodować program, tzn. postąpić dokładnie odwrotnie niż procedura odkodowująca. W naszym przypadku należałoby wykonywać.

XOR #CC
RRCA
ADD 34

W „Art Studiu", nie wiadomo po co, została umieszczona procedura kodująca. Znajduje się ona pod adresem 26003, a wygląda w ten sposób:

```
26003 21C165  LD HL,26049   ;Adres początku.
26006 11476C  LD DE,27719   ;Adres końca.
26009 7E      LD A,(HL)     ;Pobranie bajtu z
26010 EECC    XOR #CC       ;pamięci, zakodo-
26012 0F      RRCA          ;wanie go za  po-
26013 C622    ADD 34        ;mocą  XOR, RRCA,
26015 77      LD (HL),A     ;ADD i odesłanie.
26016 23      INC HL        ;Następny adres.
26017 B7      OR A          ;Sprawdzenie  wa-
26018 ED52    SBC HL,DE     ;runku końca,skok
26020 19      ADD HL,DE     ;do pętli lub po-
26021 20F2    JR NZ,26009   ;wrót (do Basica)
26022 C9      RET
```

Jak widać — jest ona zbudowana analogicznie, jak procedura odkodowująca. Jeśli takiej procedury nie ma — można ją szybko i prosto napisać na podstawie procedury odkodowującej (np. ATIC ATAC, NIGHT SHADE, THE LAST WORD).

W zabezpieczeniach programów stosowane są także mało znane, a nie publikowane w katalogach firmowych rozkazy mikroprocesora Z-80. Dzięki ich zastosowaniu działanie programu staje się mało czytelne, a i oglądanie go disasemblerem jest utrudnione

Zacznijmy od początku. Najczęściej spotykanym rodzajem instrukcji niepublikowanych są rozkazy operujące na połówkach rejestrów indeksowych **IX** i **IY** w grupie rozkazów nie poprzedzonych żadnym innym prefiksem (tzn. CBh lub EDh). Polegają one na prefiksowaniu kodem DDh lub FDh rozkazu dotyczącego rejestru H lub L. Wówczas zamiast tego rejestru brana jest odpowiednia połówka rejestru indeksowego. Przez **hX** oznacza się zwykle starszą część rejestru **IX**, przez **lX** — młodszą. Analogicznie **hY** i **lY**. Oto przykłady

kod	istrukcja				
24	INC H	DD24	INC hX	FD24	INC hY
2D	DEC L	DD2D	DEC lX	FD2D	DEC lY
4C	LD C,H	DD4C	LD C,hX	FD4C	LD C,hY
64	LD H,H	DD64	LD hX,hX	FD64	LD hY,hY
2601	LD H,1	DD2601	LD hX,1	FD2601	LD hY,1
B5	OR L	DDB5	OR lX	FDB5	OR lY

Prawdziwe jest to dla wszystkich rozkazów przesłań jednobajtowych między rejestrami i ośmiobitowych operacji AND, OR, XOR, ADD, ADC, SUB, SBC i CP — wykonywanych w akumulatorze

Prefiks FDh lub DDh dotyczy wszystkich rejestrów H, L lub HL występujących w rozkazie, a więc nie jest możliwe użycie w jednej instrukcji komórki adresowanej przez (HL), rejestru HL, H lub L równocześnie z hX, hY, lX, lY, IX lub IY, (w dalszej części ograniczymy się do rejestru IX, choć wszystko to dotyczy także rejestru IY) np

```
66 LD H,(HL)     DD66dd LD hX,(IX+d)
75 LD (HL),L     DD75dd LD (IX+d),lX
65 LD H,L        DD65   LD hX,lX
```

Nieco inaczej przedstawia się sprawa rotacji komórki adresowanej przez rejestr indeksowy, tzn. instrukcji rozpoczynających się kodem DDCB. Instrukcje typu **RR (IX + dd)** itp. są szczegółowo opisane we wszelkich dostępnych materiałach o mikroprocesorze Z-80, mało

kto wie jednak o instrukcjach typu **RR (IX+dd),r** itp., gdzie r oznacza dowolny rejestr mikroprocesora. Polegają one na prefiksowaniu kodem DDh lub FDh instrukcji typu **RR r**. Podobnie rzecz się ma z instrukcjami **SET n,(IX+d),r** oraz **RES n,(IX+d),r** (ale dla **BIT** — już nie). Działanie takiej instrukcji polega na wykonaniu „normalnego" rozkazu **RR (IX+d)** (lub podobnego), **SET n,(IX+d)** lub **RES n,(IX+d)** a następnie przesłaniu wyniku zarówno do komórki (IX+d) jak i do odpowiedniego rejestru wewnętrznego mikroprocesora. Np.:

```
CB13      RL E
DDCB0113  RL (IX+1),E
```

powoduje rotację komórki o adresie IX+1 w lewo o 1 bit i odesłanie tego wyniku do rejestru E, co w „konwencjonalny" sposób należałoby zrobić:

```
DDCB0116  RL (IX+1)
DD5E01    LD E,(IX+1)
```

Na zakończenie omawiania tego typu instrukcji należy jeszcze przypomnieć, że nie ma rozkazu EX DE,IX czy EX DE,IY. Prefiksowanie rozkazu **EX DE,HL** nie daje żadnego skutku. Podobnie — prefiksowanie rozkazów, których kody rozpoczynają się od EDh oraz tych, w których nie występuje żaden z rejestrów H, L lub pary HL (np. **LD B,n**, **RRCA** itp.)

Kolejnym ciekawym rozkazem jest **SLI** (Shift Left and Increment), którego działanie jest analogiczne do **SLA**, z tą różnicą, że najmłodszy bit jest ustawiany na 1. Zachowanie wskaźników jest identyczne, jak przy SLA lub innych obrotach.

```
CB37       SLI A
CB36       SLI (HL)
DDCBdd36   SLI (IX+d)
DDCBdd37   SLI (IX+d),A
```

Czasem nieco problemów stwarza budowa rejestru flagowego, szczególnie wtedy, gdy wykorzystywany jest on w dość nietypowy sposób, np.

```
PUSH AF
POP BC
RL C
JR NC,...
```

Jego budowa jest przedstawiona na rys. 1. Dodatkową własnością rejestru F jest to, że bity 3. i 5. (oznaczone jako F3 i F5) dokładnie odzwierciedlają stan 3. i 5. bitu wyniku ostatnio wykonywanej ośmiobitowej operacji arytmetycznej lub logicznej albo operacji **IN r,(C)** i **IN F,(C)**. Np. po wykonaniu

```
XOR A        ;zapisanie wartości 0
ADD A,15     ;wynik — %00001111
```

wskaźnik F5 zostanie skasowany, F3 — ustawiony.

Ostatnią, chyba ostatnią tajemnicą Z-80 (któż zaręczy że nie ma ich więcej?) jest rejestr odświeżania pamięci R, a dokładniej — to, że gdy po kolejnym cyklu maszynowym mikroprocesora inkrementowana jest wartość tego rejestru, to jego najstarszy bit pozostaje nienaruszony, może być więc wykorzystany do pamiętania dowolnej, oczywiście jednobitowej, informacji. Ważne jest także to, że instrukcja **LD A,R** za pomocą której może być odzyskana ta informacja, ustawia również wskaźnik S, a więc nie jest potrzebna dodatkowa instrukcja sprawdzająca wartość tego bitu.

Na zakończenie kilka krótkich, ale dość często używanych zabezpieczeń. Ich likwidowanie polega zwykle na postępowaniu odwrotnym do zabezpieczenia. Najczęściej pod tym względem eksploatowane są zmienne systemowe. Np. zmienna **DFSZ** (23659) określa ilość linii w dolnej części ekranu (potrzebną do wydrukowania komunikatu o błędzie lub wprowadzania danych). Podczas działania programu wartość ta wynosi zwykle 2, lecz wystarczy wpisać tam wartość 0, by program zawiesił się przy próbie przerwania (bo drukowany jest komunikat, na który nie ma miejsca). Jeżeli w programie znajduje się instrukcja INPUT lub CLS, to ma ona podobny skutek.

Dość częstym zabezpieczeniem, używanym nawet w bardzo poważnych programach, jest zmiana wartości zmiennej **BORDCR** (23624), która określa kolor ramki i atrybutów dolnej części ekranu. Znaczenie poszczególnych bitów przedstawia rys. 2. Zabezpieczenie polega na tym, że kolor atramentu jest taki sam, jak ramki, a więc po zatrzymaniu programu możemy odnieść wrażenie, że program się zawiesił (nie widać komunikatu). Odbezpieczenie tego jest wręcz trywialne — wystarczy wpisać **BORDER 0** (lub inny kolor).

Listing 1.

```
  10 CLEAR 65361: LET s=0
  20 FOR n=65362 TO 65395: READ
 a: POKE n,a: LET s=s+a: NEXT n
  30 IF s<>3437 THEN PRINT "Błąd
 w linii DATA": STOP
  40 SAVE "beep"CODE 65362,34
  50 DATA 88,255,3,19,0,62,221,2
29,6,8,197,17,30,0,33,112,5,205
  60 DATA 181,3,33,0,8,43,124,18
1,32,251,193,16,235,221,225,201
```

Rys. 1.

7	6	5	4	3	2	1	0
S	Z	F5	H	F3	P/V	N	C

Rys. 2.

7	6	5	4	3	2	1	0
FLASH	BRIGHT	BORDER			INK		

Kolejnym prostym zabezpieczeniem jest zmiana wartości zmiennej **ERRSP** (23613-14) lub dna stosu maszynowego (zmienna ta wskazuje właśnie dno stosu) gdzie znajduje się normalnie adres procedury obsługującej błąd BASIC-a (wywoływanej przez **RST 8**). Zmniejszenie wartości ERRSP o 2 powoduje zabezpieczenie programu przed BREAK i każdym innym błędem — program ponownie się uruchamia od miejsca, w którym wystąpił błąd. Zmiana zawartości dna stosu maszynowego lub inna zmiana wartości ERRSP może powodować zawieszanie się bądź „resetowanie" komputera w wypadku błędu.

Znanym zabezpieczeniem jest również wpisanie wartości większych od 9999 do komórek pamięci oznaczających numer linii **BASIC-a** (Pierwszy bajt starszy!). Gdy mieści się on w granicach 10000 — 16383, na listingu wygląda to np. :000 (zamiast 10000) i linie te nie dają się poprawić (EDIT), jeśli natomiast przekracza 16384 — dalsza część programu uznawana jest za nieistniejącą. Można też spotkać zabezpieczenie polegające na wpisywaniu minimalnych wartości (tzn. 1) do zmiennych **REPDEL** (23561) i **REPPER** (23562) co w efekcie utrudnia pracę z komputerem, ale do jego zlikwidowania potrzebny jest tylko szybki refleks (wpisz przez POKE-i normalne wartości: 35 i 5).

Warto jeszcze wspomnieć o zmiennej **NMIADD**, która nie jest używana ze względu na błąd w ROM-ie. Miała ona zawierać adres obsługi przerwania NMI (przyjmowanego zawsze!), błąd polega na tym, że mikroprocesor skacze tam tylko wtedy, gdy wynosi on 0. Gdyby jednak dorobić do Spectrum odpowiednią przystawkę z pamięcią EPROM (np. Bajtek 7/87), droga do zatrzymania dowolnego programu stałaby otworem.

I to już wszystko. Jeżeli zainteresował cię ten temat, a masz jakiś ciekawy problem związany z odbezpieczaniem programów, napisz do redakcji (zaznaczając na kopercie „Od środka"). Być może problemów tych jest jeszcze na tyle dużo, że powstanie dodatkowy odcinek tego cyklu.

Tomasz Surmacz
Robert Dudzik

WERYFIKATOR II

Czytelnikom korzystającym z programu "Weryfikator" z 6-ego numeru Bajtka proponuję jego nowszą wersję, znacznie ulepszoną.

Pierwsza część programu służy do wydruku na ekranie lub drukarce treści programu w języku BASIC wraz z sumami kontrolnymi każdej linii. Sumy te tworzone są oczywiście w ten sam sposób, co starszym weryfikatorem. Ta część uruchamiana jest przez **RANDOMIZE USR 65269**; numer kanału transmisji przechowuje komórka 65270. Wydruk kierowany jest na ekran — **POKE 65270,2** lub drukarkę — **POKE 65270,3**.

Część druga wykorzystuje system przerwań maskowalnych Spectrum i drukuje w prawym górnym rogu ekranu sumę kontrolną linii, w której znajduje się kursor edycji. Podgląd ten uaktywniany jest przez **RANDOMIZE USR 65519**. Cały program zajmuje obszar od adresu 65269 do 65535 i jest nierelokowalny.

Piotr Strzelczyk

```
100 CLEAR 65268: LET nrlin=256:
LET adres=65269
105 READ a$: LET l=LEN a$: IF l
=0 THEN PRINT "O.K.": STOP
110 LET sum=0
115 FOR i=1 TO l-4 STEP 2
120 LET licz=(CODE a$(i)-48-7*(
a$(i)>":"))*16+(CODE a$(i+1)-48-
7*(a$(i+1)>":"))
125 POKE adres,licz: LET sum=su
m+licz: LET adres=adres+1
130 NEXT i
135 LET sump=(CODE a$(l-2)-48-7
*(a$(l-2)>":"))*256+(CODE a$(l-1
)-48-7*(a$(l-1)>":"))*16+(CODE a
$(l)-48-7*(a$(l)>":"))
140 IF sum<>sump THEN PRINT "bl
ad w lini nr.";nrlin: BEEP 1,0:
STOP
150 LET nrlin=nrlin+1: GO TO 10
5
256 DATA "3E02CD[illegible]162A535CE5CD5
5183E20D73E58F"
257 DATA "20D73E[illegible]D73E01D7DDE1E
5DD5E02DD56849"
258 DATA "031313[illegible]130600210000D
D4E0009DD232AA"
259 DATA "187AB3[illegible]F52B2B2B2B2B2
B2B2B2BE5D1596"
260 DATA "7BCB3F[illegible]3FCB3FCB3FC63
0FE3A3802C67D1"
261 DATA "07D77B[illegible]0FC630FE3A380
2C607D73E146AC"
262 DATA "D73E00[illegible]3E0DD7E1DD214
B5C7DDDBE007AC"
263 DATA "20967C[illegible]E012090C9DDE
5F5C5D5E597D"
264 DATA "2A495CC[illegible]6E19204C23235
E23562B2B2B42D"
265 DATA "131313[illegible]E5DDE1210000D
D4E0009DD23544"
266 DATA "187AB3[illegible]F52B2B2B2B2B2
B2B2B2BE5D1596"
267 DATA "2E1C7BC[illegible]3FCB3FCB3FCB3
FC630FE3A38753"
268 DATA "02C607C[illegible]D0FF7BE60FC63
0FE3A3802C6809"
269 DATA "07CDD0[illegible]E1D1C1F1DDE1F
FC9D5265836816"
270 DATA "472640[illegible]1600EB292929E
D4B365C09EB546"
271 DATA "06081A77241C10FAD12CC
9F3ED46FBC9799"
272 DATA "C36EFFF33E3BED47ED5EF
B1872E"
513 DATA ""
```

MUZYKA ?

Komputer ZX Spectrum posiada jeden generator dźwięku 1/8 oktawy. W dodatku ma on zamontowany delikatny głośniczek z cienką membraną. Jest więc niemożliwe uzyskanie tu prawdziwej muzyki komputerowej. A jednak istnieje kilka programów przeznaczonych do tworzenia muzyki, a bardzo wiele gier posiada wręcz zdumiewającą oprawę dźwiękową.

Przypatrzmy się bliżej niektórym z nich. Na przykład „Music Composer" — umożliwia on tworzenie melodii poprzez zapis nutowy na pięciolinii. Do wyboru są nuty od pełnej do 1/16, ewentualnie z bemolem lub krzyżykiem. Pięciolinia to nie tylko główne pięć linii, lecz i po dwie linie dodane, na górze i na dole. Na raz możemy oglądać tylko jej część, złożoną z 32 znaków — nut lub odstępów.

Po pracowitym wstukaniu melodii, możemy posłuchać jej w wybranym tempie. I to już wszystko, co potrafi ten program. Trudno chyba wymagać od niego więcej — napisany jest bowiem w **Basic-u.**

Podobny, lecz nieco bardziej rozbudowany jest program „Music Typewriter". Pozwala on na tworzenie muzyki na dwóch różnych pięcioliniach i odtwarzanie obu melodii równocześnie. To współbrzmienie jest oczywiście symulowane, lecz efekty są zadowalające.

Jednak najlepsze efekty uzyskuje się stosując „Wham! The Music Box". Program ten pozwala na tworzenie muzyki o oszałamiającej jakości. Są weń z góry wpisane cztery piosenki zespołu Wham, jakość ich niewiele odbiega od muzyki otrzymywanej z syntezatora. Dobrze jest wzmacniać dźwięk (wyprowadzając go np. przez MIC), gdyż głośniczek Spectrum brzęczy bardzo cicho.

Trudno jest jednak pisać taką muzykę, sam zapis nutowy nie wystarcza.

Wiadomo, że nawet średnio dopracowana gra na Commodore jest „lepsza" niż podobna na Spectrum. Istnieją wprawdzie wyjątki (np. Turbo Esprit), lecz nieliczne. Mimo to słuchając oprawy dźwiękowej gier na Spectrum często doznajemy miłych wrażeń. Oczywiste jest, że stosując BEEP nigdy nie uzyskanoby podobnych dźwięków. Zainteresowanym tworzeniem muzyki w asemblerze odsyłam do „Bajtka" nr 1/86.

Ale — do rzeczy. Po długich i zażartych dyskusjach udało się nam wyłonić dziesiątkę gier, które — naszym zdaniem — posiadają najlepszą muzykę. Oto ona:

1. DYNAMITE DAN
2. PING PONG
3. AGENT X
4. MIKIE
5. FAIRLIGHT
6. TOP GUN
7. TURBO ESPRIT
8. B.C. BILL
9. TAPPER
10. HEARTLAND

Spectrum nie może też dać sobie rady z syntezą mowy. Istnieje, co prawda, program „LMOWA" (polski!), który mimo zajmowanych 36 KB potrafi wyraźnie wypowiedzieć jedynie niektóre cyfry. I tu znów przodują producenci gier. Ostatnio, przed rozpoczęciem gry, słychać jej tytuł lub krótki śmiech. Tak jest w GHOSTBUSTERS, NODES OF YESOD, ROBIN OF THE WOOD, E.T., czy 007. Ale np. w ROBIN OF THE WOOD na wypowiedzenie zdania „now you as Robin must run for the silver arrów" potrzebuje ok. 7 KB pamięci.

Marcin Przasnyski

ANALIZATOR

Podczas ostatnich porządków natknąłem się w swoje bibliotece na nieznaną chyba w Polsce książkę włoskiego autora Roberta Rigo „Spectrum Tool", w której odkryłem bardzo ciekawy program. Jest to procedura wykrywająca obecność sygnału na złącu EAR i odzwierciedlająca go na ekranie. Po wpisaniu programu pozostaje tylko podłączenie magnetofonu do komputera, wykonanie RUN i już można analizować każdy sygnał. Jednym z ciekawszych zadań jest analiza poziomu nagrywania programów z komputera na magnetofon i umożliwienie przez to znalezienia jego optymalnego poziomu

Tomasz Cieślik

```
1000 CLEAR 31999: GO SUB 9000  E2
1010 LET x=0: LET y=USR 32000  12
1020 FOR n=0 TO 255 STEP 2  78
1040 LET l=USR 32000: PLOT n,l  30
1050 DRAW x-n,-l+y: LET x=n  F7
1060 LET y=l: NEXT n  E2
1070 CLS : GO TO 1010  2A
9000 FOR f=32000 TO 32016: READ a: POKE f,a: NEXT f  1B
9010 DATA 243,1,0,175,62,127,219,254,203,119,32,1,12,16,245,251,201  23
9020 RETURN  63
```

TIMEX

— I 64 KOLUMNY —

Przy rozszerzaniu możliwości graficznych TIMEXA w stosunku do SPECTRUM zapomniano o tak ważnej rzeczy, jaką jest prawidłowe wykorzystanie zwiększonej rozdzielczości do drukowania tekstów w formacie dwa razy zagęszczonym.

Po przejściu komendą OUT 255,6 w tryb zwiększonej rozdzielczości, teksty drukowane na ekranie zapisywane są co prawda mniejszymi znakami, lecz w co drugiej kolumnie. Wynika to z rozłączenia głównej pamięci ekranu z tzw. drugim ekranem, o większej rozdzielczości, który powinien dopełniać ekran główny. W zasadzie nie uzyskujemy więc nic nowego — nadal w wierszu znajdują się 32 miejsca na znaki.

Jedynym wyjściem z tej sytuacji jest użycie jednej ze standardowych instrukcji — na przykład LPRINT — do drukowania tekstów w 64 kolumnach. W tym celu należy strumień 2, zwykle wychodzący na drukarkę, skierować do kanału obsługującego ekran. Poniższa procedura zrobi to wszystko za nas.

Program zajmuje pamięć od adresu 64975 do 65535 i nie jest relokowalny. Wywołanie następuje przez RANDOMIZE USR 65512. Powoduje to przesunięcie obszaru programu w BASIC-u powyżej drugiego ekranu i przedefiniowanie kanału drukarki. Od tej pory, w trybie zwiększonej rozdzielczości, teksty drukowane przez LPRINT będą ukazywały się w 64 kolumnach. Składnia instrukcji pozostaje taka, jak przy normalnym PRINT, tylko OVER i INVERSE działają globalnie i po użyciu w LPRINT należy je wyłączyć. Dodatkową zaletą jest podstawienie procedury czyszczącej ekran pod nie używany znak kontrolny CHR$ 5.

Rafał Cegieła

```
   1 DATA 64975,18
   2 DATA "D9E6C5D52A515C22F85F32FD5F21FC5FCB46C268FECB4EC272FECB56C272FECB54"
   3 DATA "5EC239FFCB66C24DFFFE18DA9EFEFEA5D2C0FF3AFF5F4F3AFE5F473E189047CD16"
   4 DATA "9B0ECB393004110020190600922845C110A0AED53865C3AFA5FCD011621FC5F78"
   5 DATA "CB6E28063E14D73E01D7CB7628063E15D73E01D73AFD5FD721FF5F342AF85F2218"
   6 DATA "516C3E4021FF5FBEDC7CFE21FE5F3E168EDC86FED1C1E1D9C932FE5F21FC5FCB99"
   7 DATA "8618D932FF5F21FC5FCB8E1BCF21FF5F360021FE5F34C921FE5F36163AFB5FFE4F"
   8 DATA "00CCD4FFCD81FFCDFE0D21FB5F35C9FE06282CFE05CA60FFFE0D284AFE08284CB8"
   9 DATA "FE09284FFE0A2852FE0B2858FE14286AFE15286EFE162850FE172856C34BFE3A38"
   10 DATA "FF5F21FF5FFE11380EFE21380FFE313810CD7CFEC34BFE3610C34BFE3620C34B18"
   11 DATA "FE3630C34BFECD7CFEC34BFE21FF5F35C34BFE21FF5F34C34BFE21FE5F34C34B02"
   12 DATA "FE21FE5F35C34BFE21FC5FCBC6CBCEC34BFE21FC5FCBCEC34BFE21FC5FCBDEC378"
   13 DATA "4BFE21FC5FCBE6C34BFE21FC5FCB9ECBAE3AFD5FFE00CA4BFECBEEC34BFE21FC69"
   14 DATA "5FCBA6CBB63AFD5FFE002802CBF6C34BFECD6B0D21004011006001001BEDB03EEA"
   15 DATA "0032FF5F32FE5F32FC5F3E1632FB5FC34BFE0618C5CD9B0E11002019EBC1C57824"
   16 DATA "D60147D5CD9B0E01002009D1CDA6FFC10578FE01C818DD3E08012000E5DBEDB08E"
   17 DATA "D12100011 9EBE1010001093DFE00C818E82AF85F22615C3AFD5FD6A5119500CD8A"
   18 DATA "100CC34BFE3E1632FB5FCD8E0278FEFF28F8FE202002CF0CC921CA5C01461ECD5D"
   19 DATA "551621CFFD22C55C3E0232FA5FC9423CAD"

9990 REM "PRINT 64" © R.Cegiela
9991 REM (przy przenoszeniu do BASIC-a zostal uzyty HeToBa z "Komputera" 12/86)
9992 CLEAR 64974
9993 READ A,S: FOR F=1 TO S
9994 READ L$: LET L=LEN L$: LET S=0: LET K=2
9995 LET A$=L$(K-1): LET B$=L$(K)
9996 LET C=(CODE A$-48-(7*(A$>"@")))*16+CODE B$-48-(7*(B$>"@"))
9997 IF K<L THEN POKE A,C: LET S=S+C: LET K=K+2: LET A=A+1: GO TO 9995
9998 IF S-256*INT (S/256)<>C THEN PRINT "BLAD W LINII ";F+1: STOP
9999 NEXT F
```

PROGRAMY SPOD NUTKI

Komponowanie własnych utworów muzycznych jest pracą dość skomplikowaną, zwłaszcza jeżeli użytkownik ma zamiar dokonać tego w BASIC Commodore 64; nieco szybciej da się to zrobić stosując dyrektywy BASIC Commodore 128; choć szczerze mówiąc BASIC nie jest tu najlepszym rozwiązaniem. Sytuację tę wykorzystało wiele firm profesjonalnych produkując programy uzupełniające tę lukę.

Programów muzycznych, podobnie jak graficznych opracowano dla Commodore bardzo wiele. Dokładny opis wszystkich — jest oczywiście niemożliwy z prostego powodu — brak nam na to miejsca. Przeważają wśród nich dwa typy główne: duże procesory muzyczne umożliwiające dowolne przetwarzanie dźwięku, zapis skomponowanych utworów i oczywiście ich odtwarzanie z wykorzystaniem do maksimum wszystkich możliwości układu 6581 SID oraz elektroniczni perkusiści powtarzający cyklicznie zadany wcześniej rytm, który użytkownik może zaprogramować i dowolnie zmieniać wedle własnych potrzeb. Jako dodatkowe oprogramowanie dostępne na rynku warto wymienić programy właściwie profesjonalne umożliwiające nam sterowanie kilku orzyłączonych do C-64 syntetyzerów poprzez interfejs MIDI oraz potężną ilość programów demonstracyjnych typu WE MUSIC, GAME MUSIC, ROB HUBRADO DEMO, itp.

Ostatnią grupę stanowią programy wyspecjalizowane typu GUITAR wspomagające naukę gry na gitarze poprzez wyświetlanie układu chwytów gitarowych, automatyczne tunery-stroiki generujące częstotliwości wzorcowe czy monitory SID ukazujące co się dzieje w poszczególnych rejestrach układu dźwiękowego podczas odtwarzania danej melodii. Pojawiło się kilka programów symulujących lepiej lub gorzej syntetyzery. Ze względu na ilość miejsca w poniższym artykule postaram się omówić pokrótce dwie pierwsze grupy oprogramowania oraz wspominane syntetyzery.

MUSIC PROCESSOR

Program wyłącznie dyskowy umożliwiający zapis, odtwarzanie i edycję utworów muzycznych. Podczas odtwarzania na ekranie ukazuje się pięciolinia, gdzie wszystkie dźwięki są reprezentowane w normalnym zapisie nutowym. Program pozwala na regulację parametrów dźwięku, korzystanie z akompaniamentu (jeden generator odtwarza zapisany wcześniej akompaniament, pozostałe dwa są do dyspozycji użytkownika) Na dyskietce zapisano kilkanaście utworów demontracyjnych.

MUSIC SHOP

Podobnie jak MUSIC PROCESSOR. Dodatkową zaletą tego programu jest możliwość wydruku na drukarce zapisanego utworu w normalnym zapisie nutowym. Wydruk przykładowy przedstawiony jest poniżej. Program dyskowy.

MUSIC CONSTRUCTION SET

Program w zasadzie dyskowy, choć widziałem jego wersję zapisaną na taśmie. Układanie utworu polega na nanoszeniu za pomocą joysticka poszczególnych nut na przesuwającą się na ekranie pięciolinię. Obsługa programu jest dość wygodna i przypomina trochę program GEOS.

RHYTM ROCKIER KAWASAKI

W zasadzie jest to już program — syntetyzer przeznaczony dla użytkowników młodej generacji. Dolną połowę ekranu zajmuje klawiatura, której klawisze są podświetlane w chwili naciśnięcia odpowiedniego klawisza na klawiaturze komputera. Górna połowa to ekran graficzny, na którym wyświetlane są rozmaite figury geometryczne. Użytkownik ma do dyspozycji rozbudowany edytor dźwięku, umożliwiający zapamiętanie np. podkładu perkusyjnego czy gitary basowej i następnie ich ciągłe odtwarzanie. Podczas odtwarzania można zmienić klawiaturę w syntetyzer i improwizować. Możliwy zapis i odczyt programów, w tym także i grafiki. Dodatkowo użytkownik może generować wiele ciekawych efektów dźwiękowych przypisanych niektórym klawiszom. Ciekawe efekty audiowizualne. Program dyskowy.

SONGWRITER

Program przypominający nieco pozytywkę. W środku ekranu umieszczona jest klawiatura, na której wciśnięcie klawisza powoduje pozostawienie jasnego punktu na przesuwającej się pod nią „taśmie"; odtwarzanie polega na automatycznym bądź ręcznym przesuwie „taśmy". Program dyskowy.

MASTER COMPOSER

Rozbudowany edytor muzyczny. W programie stosowany jest zapis nutowy. Możliwe programowanie faz ADSR, zapis i odczyt utworów, synchronizowanie generatorów między sobą, filtrowanie, itp. Program wyłącznie dyskowy.

THE MUSIC SYSTEM

Jest to również bardzo rozbudowany edytor muzyczny. Umożliwia pełne przetwarzanie dźwięku z filtrowaniem, synchronizacją, zapisem i odczytem zbiorów (są także demontracyjne). Sterowanie programem jest dość skomplikowane, co nie ułatwia pracy początkującym. Program dyskowy.

THE MUSIC-STUDIO

Bardzo przyjemny program — edytor muzyczny. Choć sterowanie nim nie należy do najprostszych (zwłaszcza przechodzenie menu — menu), to daje on w zamian pełny zapis nutowy oraz sterowanie joystickiem w trybie pracy edytora. Możliwa regulacja wszystkich parametrów. Wizualnie użytkownik przesuwa jedynie ładnie wykreślone potencjometry co daje także natychmiastową zmianę dźwięku. Na dyskietce zapisano ustawienie odpowiadające wielu instrumentom muzycznym. Program dyskowy.

MUSICALC

Chyba najbardziej rozbudowany (na 3 dyskietkach) edytor muzyczny. Umożliwia on stosowanie wielu rzadko spotykanych w innych programach możliwości — m.in. użytkownik może wybrać sobie odpowiednią skalę i gamę od najbardziej podstawowych do raczej egzotycznych i szczerze mówiąc rzadko potrzebnych. Możliwe jest także przypisywanie poszczególnych dźwięków do klawiszy wybranych przez użytkownika i wiele innych.

DIGITAL DRUMS (DIGI-DRUMS)

Jest to program bardzo ciekawy, symulujący (całkiem nieźle zresztą) elektroniczną perkusję. Aby nie ograniczyć użytkownika, jest on zapisany jako rozszerzenie języka BASIC — dzięki temu operator może stosować specjalne polecenia dające w zamian dźwięk poszczególnych instrumentów perkusyjnych. Do dyspozycji użytkownika pozostawiono ok. 12 KB pamięci, w czym można spokojnie zapisać cały podkład perkusyjny do długiego nawet utworu. Program dyskowy, choć po niewielkich przeróbkach można go przenieść na taśmę.

FUNKY DRUMMER

Także elektroniczny perkusista. Programowanie odbywa się na ekranie, gdzie każdy instrument perkusyjny ma swoją linię (przypomina to do złudzenia zapis nutowy dla perkusji). Poszczególne frazy można powtarzać w wielu kombinacjach i zapamiętywać. Dźwięk jest całkiem niezłej jakości i rzeczywiście przypomina perkusję — znam niestety takie programy, które mają z perkusją wyłącznie wspólną nazwę. Możliwy do zapisania na taśmie.

DIGITAL RECORDING STUDIO

Program ten wyraźnie odbiega od pozostałych i może służyć jako wyśmienity edytor do efektów specjalnych. Na ekranie pojawia się konsoleta mikserska (graficznie wykonana niestety dość słabo), na której możemy zmieniać parametry każdego z dźwięków. Możliwe jest wczytywanie całych sekwencji (opcja SEQUENCE) zapis, łączenie i odczyt pojedyńczych dźwięków, cofanie i przewijanie (tak jak w magnetofonach wielośladowych), zapamiętywanie, itp. Program dyskowy.

3001 A SOUND ODYSSEY

Zdecydowanie najlepszy program symulujący syntetyzer, jaki miałem okazję do tej pory spotkać. Oprócz rzeczywiście dobrej szaty graficznej i świetnego programu demonstracyjnego obrazującego możliwości programu, jest on także wyposażony w część pomocniczą, z której użytkownik może nauczyć się terminologii stosowanej w muzyce elektronicznej. Program umożliwia oczywiście sterowanie wszystkimi parametrami dźwięku, synchronizację i wzajemną modulację, zapamiętywanie całych sekwencji i pojedyńczych dźwięków, oraz wiele innych. Program dyskowy.

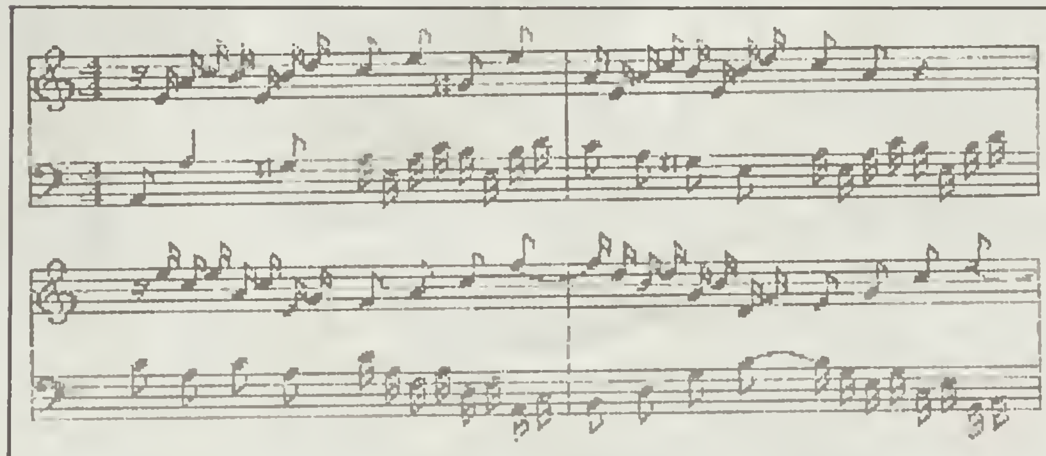
SYNTHETIZER 64

Program — syntetyzer. Moim zdaniem ustępuje on programowi 3001 A SOUND ODYSSEY. Nieco mniejsze możliwości, choć także może być to program interesujący. Zapisano w nim kilka przygotowanych wcześniej sekwencji dźwięków, sterowanie jest dość proste. Na dyskietce zapisano także kilkanaście utworów demonstracyjnych.

SYNTHIMAT 64

Program na poziomie 3001 A SOUND ODYSSEY. Można regulować wszystkie parametry dźwięku, choć sterowanie nie jest najprostsze z możliwych. Program dyskowy.

Klaudiusz Dybowski



Rys. 1 Fragment Inwencji XIII J.S. Bacha zapisanej za pomocą programu Music Shop

CHIP NA PIĘCIOLINII



Od dość dawna otrzymywałem od Czytelników listy z prośbą o publikację opisu rejestrów układu dźwiękowego Commodore 64 i 128; korzystając zatem z tematu przewodniego tego numeru „BAJTKA" wychodzę naprzeciw życzeniom Czytelników.

W obu modelach COMMODORE — C-128 i C-64 — układ dźwiękowy zajmuje dokładnie te same obszary pamięci. Posiadacze C-128 mogą oczywiście korzystać z odpowiednich poleceń języka BASIC V7.0, choć nie umożliwiają one wykorzystania wszystkich walorów układu 6581 — przykładowo programowanie dźwięku i przerwań musi odbywać się już w języku maszynowym. Posiadacze Commodore 64 muszą niestety posługiwać się instrukcją POKE, co po pierwsze wymaga większej wiedzy, a po drugie jest dosyć uciążliwe.

Układowi SID przypisano adresy pamięci 54772 do 54300, w czym zawarte są także dwa rejestry obsługujące przetworniki analogowo-cyfrowe. Ponieważ użytkownik ma do dyspozycji trzy generatory dźwięku, wypadałoby po 9 komórek na jeden generator, co jest ilością znikomą w porównaniu do możliwości tego układu. Podobnie jak w grafice wykorzystano więc do sterowania tym układem poszczególne bity tych rejestrów; ich omówienie przedstawiam poniżej. Adresy podawane na początku każdego z opisów odnoszą się do generatora nr 1 (pierwszy adres), generatora nr 2 (drugi adres) i generatora nr 3 (adres trzeci); w przeciwnym wypadku wyjaśnienie funkcji pojedynczego rejestru podane jest w samym opisie.

Komórki 54272, 54279 i 54286

Do rejestrów tych wpisuje się młodszy bajt częstotliwości dźwięku. Liczba ta musi być wartością w przedziale 0-255. Obliczanie częstotliwości dźwięku jaką chcemy uzyskać, polega na podzieleniu jej przez 0.06097. Następnie odrzucamy resztę (np. za pomocą funkcji INT) i dzielimy powstałą liczbę całkowitą (nazwijmy ją HI) przez 256. Otrzymaną wartość (jest to starszy bajt częstotliwości dźwięku) wpisujemy do komórki 54273, 54280 lub 54287 w zależności od tego z którego generatora korzystamy. Młodszy bajt otrzymujemy ze wzoru:

LO = HI — (256 × Starszy BAJT)

Komórki 54273, 54280 i 54287

Starszy bajt częstotliwości dźwięku. Zmiana wartości tu wpisanej powoduje znaczne obniżenie lub podniesienie odtwarzanego dźwięku. Im większą wartość liczbową wpisujemy do tych komórek, tym wyższy będzie uzyskiwany ton.

Komórki 54274, 54282 i 54288

Młodszy bajt szerokości (czasu trwania) impulsu, jeżeli wykorzystujemy prostokątny kształt fali (patrz komórka 54276).

Komórki 54275, 54283 i 54289

Starszy bajt szerokości impulsu (jeżeli przebieg ten wykorzystujemy). Na uwagę zasługuje fakt, że bajt ten określają tylko 4 młodsze bity tego bajtu (o numerach 0—3), natomiast bity 4—7 są nieużywane. Oba te rejestry nie mają wpływu na dźwięk, jeśli wybrano inny kształt fali niż prostokątny.

Komórki 54276, 54284 i 54290

Rejestr kontrolny. Każdy z bitów tego rejestru pełni inną funkcję. Jeżeli dany bit przyjmie stan logiczny 1, to oznacza to wybranie danej funkcji.

- Bit 7 — Kształt fali: szum (noise).
- Bit 6 — Kształt fali: prostokątny (pulse).
- Bit 5 — Kształt fali: piłokształtny (sawtooth).
- Bit 4 — Kształt fali: trójkątny (triangle).
- Bit 3 — Bit testowy, powoduje wyłączenie odpowiednio generatora nr 1, 2 lub 3.
- Bit 2 — Zastąpienie fali trójkątnej modulacją kołową generatora nr 1 z wyjściem generatora nr 3.
- Bit 1 — Synchronizacja generatora nr 1 z generatorem nr 3.
- Bit 0 — Ustawienie tego bitu powoduje rozpoczęcie faz narastania, opadania i wybrzmiewania dźwięku (fazy A, D i S), gdy jest on w stanie logicznym 0 rozpoczyna się faza zanikania dźwięku (faza R).

Należy pamiętać, że obwiednie wyjściowe (kształty fali) nie dodają się. Jeżeli wybrana została więcej niż jedna obwiednia dla danego generatora, to w wyniku otrzymamy dźwięk powstały przez przeprowadzenie na tych obwiedniach operacji logicznej AND.

W celu uzyskania modulacji kołowej należy wybrać obwiednię trójkątną i ustawić częstotliwość drugiego generatora na wartość różną od zera. Synchronizowanie i modulacja kołowa są przeprowadzane na różnych generatorach, stąd też odpowiednie sterowanie musi być wpisane do właściwego rejestru kontrolnego. W poszczególnych rejestrach są synchronizowane (modulowane) następujące generatory:

- Rejestr 54276 — Generator nr 1 z generatorem nr 2.
- Rejestr 54283 — Generator nr 2 z generatorem nr 1.
- Rejestr 54290 — Generator nr 3 z generatorem nr 2.

Komórki 54277, 54284 i 54291

Jest to rejestr kontrolny umożliwiający sterowanie czasem trwania, narastania i opadania dźwięku (fazy A i D — Attack i Decay).

- Bity 0—3 Regulacja czasu trwania fazy opadania (Decay).
- Bity 4—7 Regulacja trwania fazy narastania (Attack).

Komórki 54278, 54285 i 54292

Jest to rejestr kontrolny umożliwiający sterowanie czasem wybrzmiewania i zanikania dźwięku (fazy S i R — Sustain i Release).

- Bity 0—3 Regulacja czasu trwania fazy zaniku (Release).
- Bity 4—7 Regulacja czasu trwania fazy wybrzmiewania (Sustain).

Na tym rejestrze kończą się również funkcje układu wspólne dla wszystkich trzech generatorów.

Komórka 54293

Młodszy bajt określający częstotliwość filtra. Należy pamiętać, że wykorzystywane są w tym celu tylko trzy pierwsze bity tego rejestru — 0, 1 i 2.

Komórka 54294

Starszy bajt określający częstotliwość filtra.

Komórka 54295

Rejestr ten decyduje o rezonansie wybranego filtra i umożliwia określenie generatora, którego sygnał wyjściowy będzie poddawany filtracji.

- Bity 7—4 Rezonans filtra.
- Bit 3 — Filtrowaniu poddawany będzie sygnał zewnętrzny (wprowadzany przez linię AUDIO IN), jeżeli bit ten przyjmie stan logiczny 1.
- Bit 2 — Filtrowaniu będzie poddane wyjście generatora nr 3.

Bit 1 — Filtrowaniu będzie poddane wyjście generatora nr 2.
Bit 0 — Filtrowaniu będzie poddane wyjście generatora nr 1

Komorka 54296

Rejestr ten decyduje o głośności dźwięku oraz o wyborze jednego z trzech typow filtrow dolnoprzepustowego, gornoprzepustowego lub pasmowego

Bit 7 — Filtrowanie częstotliwością podawaną na wyjście generatora nr 3 (wybór tej funkcji jest mozliwy, gdy bit ten przyjmie stan logiczny 0).
Bit 6 — Typ filtra, górnoprzepustowy.
Bit 5 — Typ filtra pasmowy.
Bit 4 — Typ filtra dolnoprzepustowy
Bity 3—0 Głośność (natężenie) dźwięku. Im większą wartość tu wpiszemy (maksymalna = 15) tym głośniejszy będzie dźwięk

Komorki 54297 i 54298

Przetwornik analogowo-cyfrowy 1 i 2

Za ich pomocą możliwy jest odczyt popularnych swego czasu „wiosełek" (paddles).

Komórka 54299

Rejestr ten jest bardzo często wykorzystywany jako uniwersalny generator liczb losowych i bazuje on na wartościach wytwarzanych przez generator nr 3

Komórka 54230

Jest to dodatkowe wyjscie generatora nr 3 umożliwiające nam np. modulowanie wyjść pozostałych generatorów w celu otrzymania ciekawych efektów dźwiękowych, filtrowanie, itp. Należy pamiętac, że aby uzyskać jakikolwiek efekt konieczne jest uprzednie włączenie generatora nr 3.

Na zakończenie parę uwag. Jak wiadomo, układ SID umożliwia nam niezależne programowanie każdego z generatorów co oznacza, że każdy z nich może wytwarzać zupełnie inny i niezalezny od pozostałych dźwięk. Jedynie głośnośc jest ustawiana dla wszystkich trzech generatorów razem.

W wypadku C-128 istnieje mały problem, gdyż jak wiadomo komputer ten moze pracowac w oparciu o dwie częstotliwosci zegara 1 i 2 MHz, stąd tez, gdy wykorzystujemy zegar 2 MHz wzór na obliczanie częstotliwosci zmienia się — należy wtedy zastosować współczynnik 0.12159 zamiast 0.06097

Obsługa i uzyskiwanie dźwięku w trybie pracy C-64 czy wogóle na C-64 jest dosyć uciążliwa, niecierpliwym proponowałbym więc zastosowanie jednego z wymienionych obok programów umożliwiających pełne przetwarzanie dźwięku w języku maszynowym, co znacznie zwiększa szanse na uzyskanie takich brzmień czy efektów o jakie nam chodzi. Początkującym polecałbym dokładne prześledzenie i przerobienie podstawowego kursu podanego w instrukcji obsługi komputera

Dominik Falkowski

—USA KONTRA EUROPA—

W chwili, gdy to piszę wiele osób zapewne siądzie niebawem na walizkach udając się na wywczasy czy to do krajów europejskich, czy za Atlantyk. Przypuszczalnie wielu naszych rodaków zamierza wykorzystać swój wyjazd między innymi w celu zakupu jakiegoś taniego komputera 8-bitowego; takim właśnie zakupom chciałbym poświęcić ten artykuł.

Wyjeżdżający do krajów europejskich mają w zasadzie problem z głowy, gdyż komputery tam kupione nie wymagają żadnych przeróbek (może z wyjątkiem wmontowania dekodera systemu PAL do telewizora). Zgoła inaczej ma się sprawa z tymi wszystkimi, którzy znęceni znacznie niższymi cenami jadą do USA czy Kanady. **Zakup komputera w tych krajach jest zwykle nieco tańszy niż w Europie lecz są to tylko pozory.** Każdy Commodore kupowany w USA czy Kanadzie pracuje w systemie telewizyjnym NTSC (z wyjątkiem komputerow sprzedawanych w dużych centrach handlowych, gdzie może zabłąkać się niechcący turysta z Europy). Oznacza to, że jeśli użytkownik nie zakupi (lub nie posiada w domu) telewizora czy monitora, który umożliwiałby odbiór sygnałów w tym systemie, to może się od razu pożegnać z kolorami. Wiele firm elektronicznych w Polsce przestraja telewizory z systemu PAL na SECAM (stosowany w Polsce) i odwrotnie, za NTSC w zasadzie nikt nie chce się brac a jeśli już, to koszty takich przeróbek są odpowiednio wysokie. Proponowałbym również upewnienie się w sklepie w jakim systemie TV pracuje kupowany komputer.

Z NTSC wiążą się jeszcze dwie mało przyjemne sprawy. Po pierwsze komputer generuje sygnał wizyjny w zakresie fal VHF (a nie UHF jak to jest w komputerach europejskich) na kanale 3 lub 4, co odpowiada mniej więcej naszemu pierwszemu programowi TV. Dzięki temu obraz jest znacznie bardziej „miękki" i rozlany w przeciwieństwie do ostrego obrazu generowanego przez europejskie wersje Commodore. Po drugie występują tu pewne problemy związane z uruchamianiem niektórych programów komputerowych, o czym pomowię za chwilę.

Drugą niemniej ważną sprawą jest zasilanie. Wyobraźmy sobie, że przywozimy z zagranicy kompletny system — drukarkę, komputer, stację dysków i monitor. **Ponieważ, USA napięcie sieci wynosi 110V/60Hz (w Polsce 220V/50Hz), nasz zestaw będzie wymagał zastosowania solidnego transformatora o odpowiedniej mocy, co znowu wiąże się z ekstra wydatkiem czy to za granicą, czy to w kraju.**

Dotyczy to zresztą wszystkich komputerów kupowanych w USA czy Kanadzie, chyba, że użytkownik kupi od razu sprzęt o parametrach europejskich, co jest jednak nieco drozsze niż ceny podawane w reklamach (obejmujących głównie sprzęt na rynek amerykański).

Trzecia sprawa to oprogramowanie. Amerykańskie i kanadyjskie wersje Commodore pracują nieco szybciej niż ich europejskie odpowiedniki, głównie ze względu na nieco większą częstotliwość zegara, co być może jest bardzo wygodne w programach użytkowych lecz może przeszkadzać w grach. Do tego należy dodac niemożność uruchomienia niektórych programów czego główną przyczyną jest... system NTSC. Problem ten będzie występował w tych grach, w których kontrola programu w czasie jest oparta na zliczeniu linii rastra. NTSC pracuje w oparciu o 525 linii generowanych na ekranie podczas gdy systemy PAL i SECAM generują ich 625. Do odpowiednich rejestrow układu VIC wpisywana jest w wypadku NTSC maksymalnie wartość 262 natomiast w PAL i SECAM 312. Jeżeli teraz gra jest uruchamiana w momencie gdy raster osiągnie linię o numerze 312 to na wersjach amerykańskich Commodore jej uruchomienie jest niemożliwe, gdy w rejestrach tych tak wysoki numer nie pojawi się nigdy — po 262 zliczanie rozpoczyna się od zera.

Najgorszy kłopot wiąże się jednak z **serwisem. Gdy spali się jakiś drobny element, powiedzmy rezystor czy kondensator nie ma jeszcze większego problemu; kłopoty zaczynają się od bezpiecznika** (znacznie większy gabarytowo) **wzwyż i obejmują niektóre diody, tranzystory oraz układy scalone.** Przykładowo C-64 z RFN pracuje w oparciu o nieco zmieniony układ kontrolera wizji niż Commodore amerykański; stąd też, w wypadku jego uszkodzenia potrzebna będzie wersja amerykańska tego układu co wiąże się z koniecznością jej zakupu w USA. Oczywiście układ taki będzie nam również daleko trudniej zdobyć na warszawskiej giełdzie, gdzie od czasu do czasu można je kupic (przede wszystkim do komputerów europejskich). O cenie nie wspomnę.

Nie należy się również nastawiać że wszystkie pozostałe urządzenia dostępne na tamtych rynkach będą pracowały w warunkach europejskich. Dotyczy to niektórych przystawek specjalnych, modemów, interfejsów, itp. Przykładowo POWER CARTRIDGE nie chciał żadną miarą współpracować z C-64 z USA, podobnie jak niektore programy zapisane w EPROM. Oprócz tego nie zawsze można kupić sprzęt akurat przez użytkownika pożądany; na przykład w USA stacja 1570 jest dość mało znana, gdyż w rzeczywistości wersja ta została opracowana w Europie i jest (od czasu do czasu) importowana do USA oczywiście w wersji amerykańskiej.

Na zakończenie kilka uwag dotyczących cen. **Warto pamiętac, że większosc rzeczywiscie stosunkowo niskich cen w porownaniu np. do RFN oferują spółki zajmujące się sprzedażą wysyłkową.** Na taki zakup mogą więc pozwolić sobie osoby pozostające za granicą dłużej, powiedzmy 3-4 miesiące. Dodam też, że nie zawsze firmy te godzą się akceptowac czeki z podaną w ogłoszeniu ceną — cenniki zmieniają się dość często, niektóre zawierają nawet datę ważności („expired date"), a 6-8 tygodni w ciągu ktorych odbiorca ma otrzymać towar to kupa czasu.

Zanim więc zaczniesz zastanawiac się czy nie kupić Twego wymarzonego komputera własnie za oceanem, usiądź na chwilę z kalkulatorem w ręku, przelicz sobie ewentualne koszty jakie poniesiesz kupując komputer np. w USA, w RFN i na giełdzie. Po dokładnej analizie przekonasz się, ze **„taniosc" sprzętu na rynku amerykańskim dla polskiego kupca jest dość enigmatyczna i jakby nawet miejscami mglista.** W większości wypadków okaże się prawdopodobnie, że zysk wyniesie tu kilka, maksymalnie kilkadziesiąt dolarów, natomiast straty typu kolor, jakość obrazu, zasilanie przez transformator i znacznie utrudniony (a przy tym droższy) serwis zamienią go w stratę. Decyzję jednak zostawiam już Tobie

Klaudiusz Dybowski

DAS MUSIKBUCH ZUM COMMODORE 64
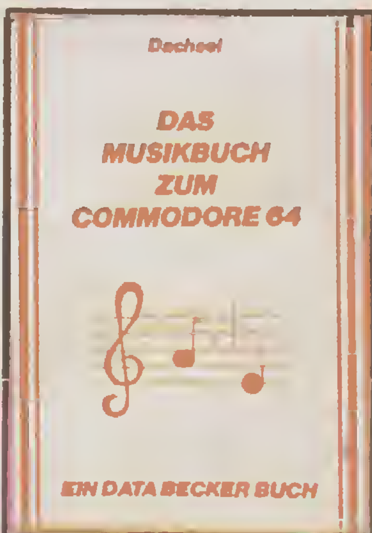
Książka ta wydana przez opisywane swego czasu w Bajtku wydawnictwo DATA BECKER jest małym kompendium wiedzy dotyczącym programowania i uzyskiwania ciekawych efektów dźwiękowych na Commodore 64 (i 128 w trybie pracy C-64). Na 208 stronach zawarto w niej informacje podstawowe dla początkujących, zasady programowania dźwięku w czasie, omówienie układu SID i jego rejestrów oraz przykłady stosowania przerwań w muzyce. Na uwagę zasługuje tu fakt, ze w jednym z załączników zamieszczono mały leksykon dotyczący terminologii muzycznej stosowanej w muzyce elektronicznej

Książka porusza także wiele problemów przeznaczonych dla bardziej zaawansowanych programistów — przykładem może być modulacja dookrężna, filtrowanie, parę słów o współpracy Commodore z syntezerami polifonicznymi. Autor książki, Dachsel był i jest miłośnikiem grupy THE BEATLES, stąd też prawie wszystkie programy przez niego proponowane, to właśnie melodie znane szeroko na całym świecie. Ma to również swoje dobre strony, gdyż możemy się bez problemów przekonać czy programowana interpretacja konkretnego utworu jest wierna i dokładna — tematy takie jak MICHELLE czy LET IT BE są wszystkim doskonale znane.

Programiści zarówno początkujący jak też i zaawansowani znajdą w tej książce wiele przykładów demonstrujących opisywane zagadnienia. Sporo listingów jest podanych w asemblerze, co może także służyć mniej wprawnym jako dobra „ściągawka" do układania własnych programów

Jak wspomniałem na początku, jest to podręcznik przeznaczony dla programistów zaawansowanych, choc omawia on zagadnienia związane z dźwiękiem od podstaw. Myslę jednak, ze i ci obecnie początkujący dojdą kiedyś do wprawy, a wtedy książka ta moze okazać się bardzo przydatna i pomocna

DASCHEL
DAS MUSIKBUCH ZUM COMMODORE 64
Wydawca: DATA BECKER GmbH
Merowingerstrasse 30
4000 Dusseldorf, West Germany
Stron 208, cena 39 DM, rok wydania 1984
ISBN 3-89011-012-6



PRZEDSTAWIAMY WARSAW BASIC (7)

PRZETWARZANIE WIELU PROGRAMÓW Z PODZIAŁEM PAMIĘCI

Rozwiązanie zagadnienia automatycznego dołączania procedur do programu głównego z pamięci zewnętrznej wymaga stworzenia mechanizmu podziału pamięci na poziomy. Poziom zerowy w tej organizacji zajmuje program główny i jego zmienne. Bezpośrednio za nim jest miejsce na poziom 1 dla procedur wywoływanych przez program główny. Poziom 2, zajmowany przez procedury wywoływane na poziomie 1, utworzy się bezpośrednio po zbiorze zmiennych poziomu 1, itd.

W przedstawianym w tym odcinku programie przewidziano możliwość utworzenia 5 poziomów. W Warsaw BASIC-u procedury można zagnieździć siedmiokrotnie (przewidziano 8 poziomów — od 0 do 7). Liczbę zagnieżdżeń ogranicza objętość stosu, na którym należy zapamiętać konfigurację pamięci operacyjnej BASIC-a poziomu, z którego się przełączamy, aby móc przypisać zmiennym systemowym TXTTAB (adresy $2B/$2C), VARTAB ($2D/$2E), ARYTAB ($2F/$30) i STREND ($31/$32) wartości określające organizację pamięci następnego poziomu. Przed przełączeniem interpreter odczytuje te wartości ze stosu.

Jeśli przełączenie następuje po raz pierwszy, to należy przed jego wykonaniem umieścić na stosie odpowiednie informacje. Służy temu procedura obsługująca słowo kluczowe £n (por. program 1, adresy $C5A1-$C5B3 oraz $C5EF-$C635). £n musi odczytać, gdzie znajduje się koniec tablic poziomu wywołującego, wstawić $00 do PW, czyli pierwszej wolnej komórki pamięci po bloku zmiennych tego poziomu, zapamiętać na stosie adres PW+1 jako wartość TXTTAB i PW+3 jako wartość VARTAB, ARYTAB oraz STREND poziomu wywoływanego. Dodatkowo £n zeruje dwie następne po PW komórki pamięci, czyli inicjuje pierwszy wskaźnik listy zawierającej tekst programu. Wyzerowanie pierwszego wskaźnika tej listy wskazuje, że w pamięci na poziomie wywoływanym nie ma programu.

Wykonanie £n przed przełączeniem będzie również konieczne zawsze wtedy, gdy objętość poziomu wywołującego zmieni się od ostatnio wykonanego przełączenia (por. program 1, adresy $C65C-$C685). Zmniejszenie lub zwiększenie objętości poziomu następuje na skutek zmian w programie na poziomie wywołującym lub na skutek utworzenia nowych zmiennych na tym poziomie. Powoduje to dezaktualizację danych o TXTTAB, VARTAB, ARYTAB i STREND poziomu wywoływanego umieszczonych na stosie.

£n pozwala wykonać w trybie bezpośrednim te działania, które interpreter Warsaw BASIC'a wykonuje automatycznie przed załadowaniem wywoływanej procedury, jeśli nie znajdzie jej w pamięci operacyjnej za poziomem wywołującym lub w RAM-dysku. Również automatycznie interpreter WB przełącza się na następny poziom i powraca zeń na poziom wywołujący po wykonaniu procedury.

W czasie wykonywania procedury przełączeń tych jest wiele wtedy, gdy używa się przekazywania parametrów przez nazwę. W zbiorze zmiennych procedury znajdują się wtedy zmienne lokalne i parametry formalne. Nazwy zmiennych lokalnych procedury mogą nie różnić się od nazw zmiennych lokalnych w innych segmentach, a mimo to ich wartości mogą być różne. Parametry formalne w polu zmiennych procedury przechowują tylko nazwy zmiennych, tablic i funkcji z segmentu wywołującego. Aby pobrać lub przypisać wartości parametrom formalnym interpreter WB przełącza się na poziom wywołujący, gdzie odszukuje odpowiednie zmienne, tablice lub funkcje będące parametrami aktualnymi odpowiadającymi odpowiednim parametrom formalnym.

Za pomocą programu 1 przełączenia poziomów można wykonywać w trybie bezpośrednim. Służą temu procedury obsługujące słowa kluczowe £p (por. program 1, adresy $C57D-$C59E oraz $C638-$C65B) i £o (adresy $C5B6-$C5EC oraz $C68B-$C6A9). £p przełącza na poziom następny, £o umożliwia powrót.

Dokonanie podziału pamięci w trybie bezpośrednim pozwala umieścić w pamięci operacyjnej jednocześnie wiele niezależnych programów. Np., jeśli w pamięci komputera znajduje się program, to aby załadować skorowidz dyskietki, należałoby program ten najpierw zapisać w pamięci zewnętrznej, aby załadowanie skorowidza go nie zniszczyło. Użytkownik C-64, który cierpliwie rozbudowuje interpreter w oparciu o publikowane w kolejnych odcinkach tego cyklu programy, może w tej sytuacji odstąpić od nagrywania. Rozbudowany interpreter pozwala mu utworzyć nowy poziom (£n), przełączyć się nań (£p), załadować katalog (LOAD"$",8), wyświetlić go (LIST), a po obejrzeniu powrócić do programu na poziomie 0 (£o). Numer bieżącego poziomu sygnalizuje komenda £m.

Czytelnicy, którzy „wpalcują" program 2, dołączą go do poprzednich części i uruchomią całość, mogą teraz, po wykonaniu dowolnego programu na poziomie 0, utworzyć poziom 1 i przełączyć się nań (£n £p), załadować lub wprowadzić z klawiatury drugi program, wykonać go i po ponownym wywołaniu £n przełączyć się na poziom 2. Tutaj również można umieścić i wykonać dowolny program. £m pokaże ile jeszcze pozostało wolnego miejsca. Jeżeli jest miejsce, to można powtórzyć poprzednie czynności. W ten sposób możemy dojść na poziom 4, tym samym mieć w pamięci 5 programów. Próba przejścia na poziom 5 zakończy się sygnalizacją błędu ILLEGAL QUANTITY (por. program 1, adresy $C589-$C58D oraz $C575-$C57A). Po wykonaniu programu na poziomie 4 można powrócić (£o) i wykonać powtórnie dowolny program spośród wcześniej umieszczonych na poziomach od 3 do 0. Teraz przy przechodzeniu od programu na poziomie o numerze niższym na poziom wyższy wystarczy używać £p.

Czynności związane z przechodzeniem od wykonania jednego z programów do wykonania dowolnego drugiego mogą być wielokrotnie powtarzane pod warunkiem niewprowadzenia zmian do załadowanych programów. Wyjątek stanowi zawsze program umieszczony na ostatnim z wykorzystywanych poziomów.

W Warsaw BASIC-u do podobnych celów służą komendy NEW PROC (£n), PROC (£p), PROC OFF (£o) i dodatkowo OFF, która umożliwia powrót z dowolnego poziomu (od 1 do 7) bezpośrednio na poziom 0.

Ze względu na dydaktyczne walory wykonywania przełączeń w trybie bezpośrednim w następnym odcinku pokażemy zastosowanie £n, £p, £o do symulacji wywoływania procedur. Symulacja w trybie bezpośrednim będzie wstępem do zrozumienia organizacji i zaprogramowania automatycznego w trybie programowym.

Krzysztof Gajewski
Bogusław Radziszewski

Program 1

```
.,  C573 B9 C6
.,  C575 DEC $C4CA
.,  C578 LDX #$0E
.,  C57A JMP ($0300)

.,  C57D JSR $C65C
.,  C580 JSR $C638
.,  C583 INC $C4CA
.,  C586 LDA $C4CA
.,  C589 CMP #$05
.,  C58B BCC $C590
.,  C58D JMP $C575
.,  C590 JSR $C64A
.,  C593 LDA ($5F),Y
.,  C595 STA $2B,X
.,  C597 INX
.,  C598 INY
.,  C599 CPX $CFFB
.,  C59C BNE $C593
.,  C59E JMP ($C573)

.,  C5A1 LDA #$08
.,  C5A3 JSR $C67B
.,  C5A6 JSR $C638
.,  C5A9 INC $C4CA
.,  C5AC LDA $C4CA
.,  C5AF CMP #$05
.,  C5B1 BCC $C5EF
.,  C5B3 JMP $C575

.,  C5B6 RTS
.,  C5B7 LDA $C4CA
.,  C5BA BEQ $C5B6
.,  C5BC JSR $C69D
.,  C5BF JSR $C68B
.,  C5C2 JSR $C638
.,  C5C5 LDA $CFFC
.,  C5C8 BEQ $C5EC
.,  C5CA PHA
.,  C5CB LDA $CFFB
.,  C5CE CMP #$08
.,  C5D0 BEQ $C5D9
.,  C5D2 PLA
.,  C5D3 SEC
.,  C5D4 SBC #$06
.,  C5D6 CLC
.,  C5D7 BCC $C5DA
.,  C5D9 PLA
.,  C5DA SEC
.,  C5DB SBC #$01
.,  C5DD TAY
.,  C5DE LDX $CFFB
.,  C5E1 LDA ($5F),Y
.,  C5E3 STA $2A,X
.,  C5E5 DEY
.,  C5E6 DEX
.,  C5E7 BNE $C5E1
.,  C5E9 DEC $C4CA
.,  C5EC JMP ($C573)

.,  C5EF LDY #$00
.,  C5F1 TYA
.,  C5F2 STA ($31),Y
.,  C5F4 LDA $32
.,  C5F6 STA $0282
.,  C5F9 STA $60
.,  C5FB LDA $31
.,  C5FD STA $0281
.,  C600 CLC
.,  C601 ADC #$01
.,  C603 BCC $C607
.,  C605 INC $60
.,  C607 STA $5F
.,  C609 STA $2B
.,  C60B LDA $60
.,  C60D STA $2C
.,  C60F LDA $CFFB
.,  C612 CMP #$08
.,  C614 BNE $C62A
.,  C616 LDA $5F
.,  C618 CLC
.,  C619 ADC #$02
.,  C61B BCC $C61F
.,  C61D INC $60
.,  C61F STA $5F
.,  C621 LDY $60
.,  C623 STY $2E
.,  C625 STA $2D
.,  C627 JSR $C6A1
.,  C62A LDA #$00
.,  C62C TAY
.,  C62D STA ($2B),Y
.,  C62F INY
.,  C630 STA ($2B),Y
.,  C632 JSR $C5BF
.,  C635 JMP ($C573)

.,  C638 JSR $C64A
.,  C63B STY $CFFC
.,  C63E LDA $2B,X
.,  C640 STA ($5F),Y
.,  C642 INX
.,  C643 INY
.,  C644 CPX $CFFB
.,  C647 BNE $C63E
.,  C649 RTS

.,  C64A LDA $C4CA
.,  C64D ASL
.,  C64E ASL
.,  C64F ASL
.,  C650 TAY
.,  C651 LDX #$02
.,  C653 LDA $CFFC,X
.,  C656 STA $5E,X
.,  C658 DEX
.,  C659 BNE $C653
.,  C65B RTS

.,  C65C LDY $C4CA
.,  C65F LDA $CFF1,Y
.,  C662 CMP #$08
.,  C664 BNE $C686
.,  C666 INY
.,  C667 TYA
.,  C668 JSR $C64D
.,  C66B DEY
.,  C66C LDA ($5F),Y
.,  C66E CMP $32
.,  C670 BNE $C686
.,  C672 DEY
.,  C673 LDA ($5F),Y
.,  C675 CMP $31
.,  C677 BNE $C686
.,  C679 LDA #$08
.,  C67B STA $CFFB
.,  C67E LDY $C4CA
.,  C681 INY
.,  C682 STA $CFF0,Y
.,  C685 RTS

.,  C686 LDX #$15
.,  C688 JMP ($0300)

.,  C68B LDY $C4CA
.,  C68E LDA $CFF0,Y
.,  C691 STA $CFFB
.,  C694 RTS

.,  C695 LDA $37
.,  C697 LDY $38
.,  C699 STA $33
.,  C69B STY $34
.,  C69D LDA $2D
.,  C69F LDY $2E
.,  C6A1 STA $2F
.,  C6A3 STY $30
.,  C6A5 STA $31
.,  C6A7 STY $32
.,  C6A9 RTS

.,  C45B A0 C5
.,  C45D B6 C5
.,  C45F 7C C5
.,  CFFD A0 CF
```

Program 2

```
400 PRINT"Część 4
402 X=50547: N=310: C=0
404 FOR I=0 TO N: READA: POKEX+I,A: C=C+A: NEXT
406 IF C<>41967 THEN PRINT"Błąd w części 4":END
408 PRINT"Część 4 OK"
410 Y=50267: Z=197: POKEY+1,Z: POKEY+3,Z: POKEY+5,Z
412 POKEY,160: POKEY+2,182: POKEY+4,124
414 Z=53248: POKEZ-3,160: POKEZ-2,207
416 DATA 182,197,206,202,196,162,14,108,0,3,32,92
418 DATA 198,32,56,198,238,202,196,173,202,196,201,5
420 DATA 144,3,76,117,197,32,74,198,177,95,149,43
422 DATA 232,200,236,251,207,208,245,108,115,197,169,8
424 DATA 32,123,198,32,56,198,238,202,196,173,202,196
426 DATA 201,5,144,60,76,117,197,96,173,202,196,240
428 DATA 250,32,157,198,32,139,198,32,56,198,173,252
430 DATA 207,240,34,72,173,251,207,201,8,240,7,104
432 DATA 56,233,6,24,144,1,104,56,233,1,168,174
434 DATA 251,207,177,95,149,42,136,202,208,248,206,202
436 DATA 196,108,115,197,160,0,152,145,49,165,50,141
438 DATA 130,2,133,96,165,49,141,129,2,24,105,1
440 DATA 144,2,230,96,133,95,133,43,165,96,133,44
442 DATA 173,251,207,201,8,208,20,165,95,24,105,2
444 DATA 144,2,230,96,133,95,164,96,132,46,133,45
446 DATA 32,161,198,169,0,168,145,43,200,145,43,32
448 DATA 191,197,108,115,197,32,74,198,140,252,207,181
450 DATA 43,145,95,232,200,236,251,207,208,245,96,173
452 DATA 202,196,10,10,10,168,162,2,189,252,207,149
454 DATA 94,202,208,248,96,172,202,196,185,241,207,201
456 DATA 8,208,32,200,152,32,77,198,136,177,95,197
458 DATA 50,208,20,136,177,95,197,49,208,13,169,8
460 DATA 141,251,207,172,202,196,200,153,240,207,96,162
462 DATA 21,108,0,3,172,202,196,185,240,207,141,251
464 DATA 207,96,165,55,164,56,133,51,132,52,165,45
466 DATA 164,46,133,47,132,48,133,49,132,50,96
```

PROGRAM — FORMA I TREŚĆ

Użytkownicy sprzętu i oprogramowania profesjonalnego dawno już spostrzegli, że kierunek rozwoju rynku informatycznego charakteryzuje się coraz większymi „ustępstwami" na rzecz potencjalnego nabywcy.

Pojawia się coraz więcej programów, nie wymagających szczególnie specjalistycznej wiedzy i studiowania grubych instrukcji. Programy tworzone są w taki sposób, by umozliwiały pracę nieomalże od razu, podając alternatywnie wybierane polecenia w jednoznacznej, zrozumiałej dla użytkownika formie. W konsekwencji zasadnicza treść programu staje się tylko niewielkim fragmentem „przyjaznej dla uzytkownika" konstrukcji, ponieważ forma zaczyna decydować o konkurencyjności i walorach użytkowych. Staje się więc równie ważna jak treść.

Poniższy program, nie pretendując oczywiście do czołówki światowej, ma za zadanie zilustrowanie przewagi formy nad treścią, ale przewagi koniecznej dla zapewnienia minimum komfortu użytkownikowi i jednoznaczności w sposobie prezentacji wyników. Przy konstrukcji programu posłużyłem się przykładem często występujących w elektronice obliczeń wartości rezystorów, które w odpowiedniej konfiguracji połączeń mają za zadanie osłabić określoną ilość razy przychodzący sygnał elektryczny. W dodatku układ taki, zwany tłumikiem, w pewnych przypadkach powinien charakteryzować się jeszcze jedną cechą: nie może, poza osłabieniem sygnału, naruszać swoistej równowagi w obwodzie elektrycznym zwanej dopasowaniem.

Przykładem dopasowania jest każda instalacja antenowa: przewód antenowy musi być dopasowany do anteny i do gniazdka w odbiorniku telewizyjnym. Jeżeli w pewnych sytuacjach sygnał ze stacji jest zbyt silny, to można go osłabić korzystając z jednego z typów tłumików prezentowanych i obliczanych w programie. Wybór tłumika zależy od rodzaju przewodu antenowego: dla przewodu płaskiego (280) wybieramy konstrukcje symetryczne, a dla okrągłego (75) — konstrukcje asymetryczne.

Oczywiście mozna korzystać z programu do obliczania tłumików przeznaczonych do dowolnych zastosowań i innych zakresów rezystancji. Na 56 linii programu jedynie cztery realizują obliczenia i 2 służą do komunikacji z użytkownikiem. Pozostałe linie (poza komentarzami ułatwiającymi zrozumienie organizacji programu), a więc aż 40!, są „nadbudową" służącą do kształtowania formy, czyli graficznej prezentacji wyników. Wprawdzie mozna się jeszcze pokusić o pewne skrócenie „nadbudowy", ale preferując czytelność programu, zostawiam to zadanie pilniejszym Czytelnikom „Bajtka".

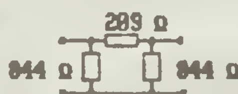
W liniach 300, 320, 340 i 360 zdefiniowano podstawowe elementy graficzne potrzebne do konstrukcji grafiki. W linii 10 zdefiniowane zostały tablice współrzędnych ekranu graficznego x i y, oraz tablice pomocniczych wskaźników (g) skoków do podprogramów definiujących elementy i wskazników wydruku elementów (r). Korzystając z mozliwości definiowania początku układu współrzędnych x, y i wskaźniki g, rysując odpowiednie elementy w odpowiednim miejscu ekranu. Po wprowadzeniu przez użytkownika informacji (linia 130 i 140) program oblicza wartości rezystorów, nanoszone następnie w odpowiednich miejscach schematu połączeń przez pętlę w liniach 200 do 280. Wykorzystując te same tablice x, y wczytywane są współrzędne lokalizacji wartości rezystorów oraz wskaźnika, wyświetlane na ekranie w przewidzianym miejscu, dzięki poleceniu TAG w linii 180 (możliwość lokalizacji tekstu z precyzją rozdzielczości ekranu graficznego).

Niewtajemniczonym winien jeszcze jestem wyjaśnienie tajemniczych jednostek tłumienia — decybeli, w skrócie dB, powszechnie stosowanych w elektronice również dla określania wzmocnienia. W naszym programie jest to 20 logarytmów ze stosunku dwóch napięć — wejściowego i wyjściowego. Nie trzeba wgłębiać się w skomplikowane obliczenia, by korzystać z programu. Wystarczy zapamiętać, ze 6dB odpowiada tłumieniu =2, 10dB =3, 12dB =4, 14dB =5, 20dB =10, 34dB =50 i 40dB =100 razy. Potrzebna będzie jednak forma, by przebrnąć przez treść...

Wojciech Ziółek

```
10 CLS:MODE 2:DIM x(31),y(31), [2177]
g(31),r(19)
20 'Tworzenie rysunku            [1742]
30 FOR i=1 TO 31:READ x(i),y(i [1697]
),g(i)
40 ORIGIN x(i),y(i)             [248]
50 IF g(i)=1 THEN GOSUB 300     [1756]
60 IF g(i)=2 THEN GOSUB 320     [1740]
70 IF g(i)=3 THEN GOSUB 340     [1641]
80 IF g(i)=4 THEN GOSUB 360     [927]
90 NEXT i                       [375]
100 ORIGIN 439,222:DRAWR 36,34 [1259]
110 ORIGIN 440,256:DRAWR 37,-3 [2063]
4:ORIGIN 0,0
120 'Obliczanie rezystorow      [2094]
130 PRINT "Jaka rezystancja pr [3889]
acy (";CHR$(191);") ";:INPUT r
char
140 INPUT "Jakie ma byc tlumie [2237]
nie (dB) ? ",tlum
150 ne=2.7183:w=tlum/17.4:a1=n [2669]
e^w:a2=1/a1:kor1=(a1-a2)/(a1+a
2)
160 kor2=1/kor1:w3=tlum/8.7:a3 [5066]
=ne^w3:a4=1/a3:kor3=(a3-a4)/2:
kor4=1/kor3
170 r1=CINT(rchar*kor1):r2=CIN [5902]
T(rchar*kor2):r3=CINT(rchar*ko
r3):r4=CINT(rchar*kor4)
180 r12=CINT(rchar*kor1/2):r32 [2859]
=CINT(rchar*kor3/2):TAG
190 'Nanoszenie opisu           [1447]
200 RESTORE 500:FOR i=1 TO 19: [3072]
READ x(i),y(i),r(i)
210 MOVE x(i),y(i)              [636]
220 IF r(i)=1 THEN PRINT r1;    [1052]
230 IF r(i)=2 THEN PRINT r2;    [1235]
240 IF r(i)=3 THEN PRINT r3;    [1088]
250 IF r(i)=4 THEN PRINT r4;    [1405]
260 IF r(i)=12 THEN PRINT r12; [683]
270 IF r(i)=32 THEN PRINT r32; [2298]
280 PRINT CHR$(191);:NEXT i:TA [3023]
GOFF:LOCATE 1,23:END
290 'Rezystor poziomy           [1664]
300 DRAWR 0,2:DRAWR 2,0:DRAWR  [11605]
0,-4:DRAWR -2,0:DRAWR 0,2:MOVE
 2,0:DRAWR 8,0:DRAWR 0,5:DRAWR
 20,0:DRAWR 0,-10:DRAWR -20,0:
DRAWR 0,5:MOVER 20,0:DRAWR 8,0
:DRAWR 0,2:DRAWR 2,0:DRAWR 0,-
4:DRAWR -2,0:DRAWR 0,2:MOVE 2,
0:RETURN
310 'Rezystor pionowy           [1871]
320 DRAWR 2,0:DRAWR 0,-4:DRAWR [12373]
-2,0:DRAWR 0,4:MOVE 1,-4:DRAW
R 0,-8:DRAWR 5,0:DRAWR 0,-20:D
RAWR -10,0:DRAWR 0,20:DRAWR 5,
0:MOVE 1,-32:DRAWR 0,-8:DRAWR
1,0:DRAWR 0,-4:DRAWR -2,0:DRAW
R 0,4:DRAWR 1,0:RETURN
330 'Linia pozioma dluga        [1827]
340 DRAWR 0,2:DRAWR 2,0:DRAWR  [6916]
0,-4:DRAWR -2,0:DRAWR 0,2:MOVE
 2,0:DRAWR 74,0:DRAWR 0,2:DRAW
R 2,0:DRAWR 0,-4:DRAWR -2,0:DR
AWR 0,2:RETURN
350 'Linia pozioma krotka       [1456]
360 DRAWR 0,2:DRAWR 2,0:DRAWR  [6655]
0,-4:DRAWR -2,0:DRAWR 0,2:MOVE
 2,0:DRAWR 16,0:DRAWR 0,2:DRAW
R 2,0:DRAWR 0,-4:DRAWR -2,0:DR
AWR 0,2:RETURN
370 'Dane rysunku               [1232]
380 DATA 100,300,4,118,300,1,1 [1871]
56,300,4
390 DATA 118,302,2,156,302,2,1 [1691]
00,260,3
400 DATA 100,180,4,118,180,1,1 [1814]
56,180,4
410 DATA 118,182,2,156,182,2,1 [780]
00,140,4
420 DATA 118,140,1,156,140,4,2 [985]
60,300,1
430 DATA 298,300,1,298,302,2,2 [1060]
60,260,3
440 DATA 260,180,1,298,180,1,2 [1497]
98,182,2
450 DATA 260,140,1,298,140,1,4 [2104]
20,300,4
460 DATA 438,300,1,476,300,4,4 [1706]
38,302,2
470 DATA 476,302,2,420,220,4,4 [1783]
38,220,1
480 DATA 476,220,4              [964]
490 'Dane opisu elementow       [2054]
500 DATA 120,320,3,250,320,1,3 [1094]
06,320,1
510 DATA 440,320,1, 60,285,2,1 [1589]
65,285,2
520 DATA 240,285,4,380,285,2,4 [1386]
85,285,2
530 DATA 120,200,32,250,200,12 [1083]
,306,200
540 DATA 12,440,212,1,60,167,2 [2086]
,165,167
550 DATA 2,240,167,4,120,132,3 [1713]
2,250,132
560 DATA 12,306,132,12          [777]
```

```
Jaka rezystancja pracy (Ω) ? 280
Jakie ma byc tlumienie (dB) ? 6
```











MIEJSKA ES



lotnisko

furgon

parking

klub sportow

ciemna aleja

cmentarz

dębowa aleja

biura

szpital

park miejski

huśtawki

1. JEDZENIE
2. PARASOL
3. LIST
4. MLEKO
5. UBIÓR LOTNICZY
6. SZAL
7. PUŁAPKA NA SZCZURY
8. KLUCZYK
9. BIAŁY FARTUCH
10. NOŻYCZKI
11. KOMBINEZON
12. KLUCZ
13. TECZKA URZĘDOWA

← DROGA JEDNOKIERUNKOWA

Bajtek

URBAN UPSTART

Oto polska wersja, dość starej już i pewnie trochę zapomnianej gry „Urban Upstart". Uważam jednak, iż pojawienie się polskiej wersji spowoduje powrót „Miejskiej Eskapady" do łask.

Gra ta jest w sumie bardzo prosta i nie pochłania tak ogromnych ilości czasu jak choćby „Sherlock" czy „Hobbit" (nie mówiąc już o „Władcy Pierścieni"), a jednocześnie ciekawa.

Rozpoczyna ją krótkie wprowadzenie w czas i miejsce akcji. Znajdujemy się w małym miasteczku, do którego co roku przyjeżdżają setki turystów. Szkopuł w tym, że łatwo jest dostać się do miasta, lecz nieporównywalnie trudniej jest się z niego wydostać. No i oczywiście naszym zadaniem jest uciec z tej mieściny wszelkimi dostępnymi metodami. Jest piękny, słoneczny poranek i... rozpoczynamy zabawę.

Na początek znajdujemy się na piętrze dużego, opuszczonego domu. Pierwszą rzeczą, którą należy zrobić, to założyć leżący na podłodze kombinezon (niestety nie jest powiedziane jak wyglądamy przy rozpoczęciu gry, więc równie dobrze możemy być w garniturze, jak i w samych kąpielówkach). Po ubraniu się schodzimy na parter. Tu znajduje się główna brama. Klucz do niej leży w salonie. W domu można znaleźć jeszcze nożyczki (w spiżarni), a także butelkę płynu (w kuchennej szafce). Pierwsza z tych rzeczy jest raczej nieprzydatna, druga natomiast ma dość ciekawe „efekty uboczne", więc lepiej też ją zostawić na swoim miejscu...

Zabieramy klucz i wychodzimy na ulicę (trzeba pamiętać, że drzwi najpierw się „odklucza", a potem otwiera). Jesteśmy więc na ulicy (polska nazwa to ulica Zielona). Przed nami stoi księgarnia. Wchodzimy do niej i zabieramy książkę. Po wyjściu, kierujemy się na przystanek autobusowy, gdzie znajdziemy parasol (a dla ciekawskich nawet numer „zegarynki"). Z przystanku

jest tylko kilka kroków do śmiet
nim karta kredytowa i numer do
niego dowiemy się jaki jest numer
nkowego.

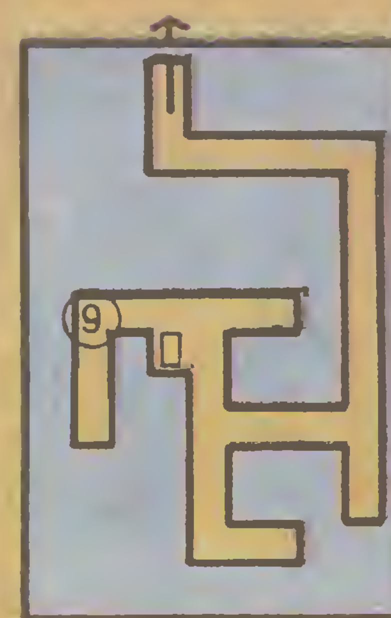
Teraz kierujemy się do budki
znamy już numer konta, więc szy
bieramy monetę i wychodzimy na u
ciężarówka. Lepiej zbytnio nie og
policja nie lubi ciekawskich. Idzie
pierwszą przecznicą (ponura Aleja)
chód. Po drodze mijamy sklep ryb
znaleźć śledzia, ale powoduje on je
wszystkich okolicznych kotów). Id
ulicy Deszczowej. Tu otwieramy par
bawić się zapalenia płuc) i idziemy
Obok kanału leży kluczyk. Zabieram
parking. Tu możemy napić się mle
krzepić!). Chwila odpoczynku i rusz
dnie ulicą Deszczową, a potem na z
tową. Mijamy szpital i wchodzimy
studni leży teczka urzędowa. Przyda
lości, więc zabieramy ją ze sobą
I teraz prosto do urzędu miejskiego
chą, niezauważeni dostajemy się
damy papiery (jak się później okaże

SKAPADA

nie tedy droga!
pustkowie
telefon
kanał
więzienie
ciężarówka
sklep rybny
bank
żółta ściana
księgarnia
urząd miejski
ul. zielona
śmietnik









ka. Tu leży list, a w
naszego kumpla, od
r naszego konta ba-

elefonicznej. Dobra,
ko do banku. Tu po-
ulicę. W pobliżu stoi
glądać, bo tutejsza
iemy na południe i
a) skręcamy na za-
ybny (można w nim
jedynie ściągnięcie
dziemy dalej, aż do
arasol (żeby nie na-
y prosto na północ.
amy go i idziemy na
leka (trzeba się po-
azamy dalej na połu-
zachód ulicą Spor-
y na górę. Tu obok
da się ona w przysz-
. Schodzimy w dół
go. Z teczką pod pa-
do środka i wykra
że, uprawniające do

wynajęcia samolotu). Teraz do „miejskiego centrum kultury" czyli do parku i pod kościołem raczymy się odrobiną jedzenia. Pokrzepieni, z nowymi siłami, kierujemy się na budowę. Tam wśród rur ukryty jest ubiór lotniczy. Zabieramy go i zakładamy. Teraz wyglądamy jak prawdziwy pilot.

Szybko, by nie ugrzęznąć w błocie, dostajemy się na pobliskie lotnisko, dajemy oficerowi dyżurnemu papiery oraz płacimy za przelot i wychodzimy na pas startowy. Wchodzimy do samolotu, tu czytamy zabraną uprzednio książkę (jest o nauce latania). Wkładamy kluczyk do stacyjki i... wreszcie jesteśmy wolni!

Ale to nie wszystko, przyda się parę dobrych rad, które umożliwią ukonozenie gry:

1 — „nie śmiecić na ulicach!" — tutejsze władze mają bzika na tle czystości (to zresztą bardzo dobrze).

2 — nie radzę interesować się zbytnio porzuconymi samochodami, zresztą nimi i tak się z tego miasta nie ucieknie.

3 — łatwo dostać w tym mieście po głowie, gdyż ścierają się tutaj dwie drużyny, których kibice są z grubsza biorąc nieodpowiedzialni...

Proponuję więc nie ubierać się w barwy żadnej z drużyn, a także nie zbliżać się do kibica w Ciemnej Alei.

4 — jeśli dostaniemy się do szpitala (co jest bardzo prawdopodobne) to musimy przebrać się za lekarza (biały fartuch), gdyż inaczej nigdy już ze szpitala nie wyjdziemy...

5 — pobyt w więzieniu (dość częsty w tej grze) możemy spożytkować na przeczytanie książki. Proponuję też być uprzejmym i cierpliwym, gdyż tamtejszy sierżant to typowy służbista. Wyjście z więzienia jest prostsze niż by mogło się na początku wydawać. Trzeba tylko trochę poczekać. Jeśli to nie daje skutku to należy czekać, aż wreszcie zadzwoni telefon i uda nam się zwiać...

Po skończeniu gry mamy możliwość rozpoczęcia jej na nowo, ale jest to zadanie o wiele trudniejsze (trudno przejść ulicę bez wylądowania w więzieniu...)

Na koniec podpowiedź: w angielskiej wersji gry wystarczy wpisać „•551117" aby zatrzymać program komunikatem STOP. Wtedy można dokonać dowolnych modyfikacji programu i nagrać go poprzez GO TO 9999.

Autor oryginału: Richard Shepherd
Komputer: ZX Spectrum 48/+

Adam Niewęgłowski

10

BAJTKOWA LISTA PRZEBOJÓW (8/88)

SPLITTING IMAGES nadal trzyma się w czołówce, walcząc z nacierającymi symulatorami lotu. Niespodziewanie pojawiły się cztery nowości — Secret Diary of Adrian Mole (znany także z telewizji i literatury), Renegade — hit stycznia 1988 roku, trzecia już część Drogi Eksplodującej Pięści i trochę już zapomniany, ale wciąż dobry Stealth.

Na to notowanie otrzymaliśmy aż 4250 propozycji, głosy uzyskały 204 gry.

1	SPLITTING IMAGES			×	×
2	ACE			×	×
3	SOLO FLIGHT	×		×	
4	HARD BALL				
5	TURBO ESPRIT		×	×	×
6	GHOST CHASER	×		×	
7	SECRET DIARY				×
8	R ENEGADE		×		×
9	EXPLODING FIST III			×	
10	STEALTH	×			

Kopertę nadesłał Zbyszek Ostrowski z Warszawy

BAJTEK
ul. Wspólna 61
00-687 Warszawa
LISTA PRZEBOJÓW

DETECTIVE

Na wiosnę 1874 roku cały Londyn był podniecony, a elegancki świat wstrząśnięty, nadzwyczaj tajemniczym morderstwem popełnionym na czcigodnym Angusie McFungus.

Angus McFungus byl drugim synem hrabiego Maynooth, ongiś gubernatora jednej z australijskich kolonii. Jego matka wróciła z Australii, by poddać się operacji katarakty. Zamieszkiwali razem przy Park Lane numer 427. Sir McFungus obracał się w najlepszym towarzystwie. Po śmierci matki nie zmienił miejsca zamieszkania. O ile wiadomo, nie miał żadnych wrogów. Życie jego, z natury cichego i beznamiętnego, płynęło spokojnie w małym kółku znajomych. A jednak na tego beztroskiego arystokratę między godziną 10 a 11 wieczór 21 marca spadła tajemnicza i niespodziewana śmierć. Dodać należy, że tego dnia skończył 72 lata.

Tragicznego dla siebie wieczoru McFungus wrócił do domu punktualnie o siódmej. Zjadł obiad i poszedł do swego gabinetu. Około jedenastej pokojówka wchodząc tam, zobaczyła go leżącego na podłodze.

Po szczególnym zbadaniu okoliczności zbrodni policja doszła do następujących wniosków:
— zbrodnię popełniła któraś z osób przebywających w domu,
— McFungus został zabity nożem,
— nie zginęły żadne papiery ani pieniądze,
— nie odnaleziono testamentu zmarłego,
— nie odkryto motywu zabójstwa.

W czasie morderstwa w domu przebywali:

Wielebny Wrinklebum — miejscowy proboszcz, jego żona zniknęła w tajemniczych okolicznościach.

Bentley — stary sługa pana McFungusa.

Hilda Crumble — powszechnie znany jest jej makaron z serem.

Gabriel — dziewczyna miła i posłuszna, lecz zupełnie bez poczucia humoru.

Major Sludgebucket — major w stanie spoczynku. Lubi wszystkim dokuczać, ale w sumie jest nieszkodliwy.

Cynthia Sludgebucket — jedyna córka majora, arogancka i zuchwała.

Profesor Bull — stary przyjaciel McFungusa. Roztargniony, ma fioła na punkcie książek.

Mr Dingle — adwokat sir McFungusa. Całkowicie lojalny.

Doktor Mortem — krótkowzroczny i trochę łatwowierny. Od lat jest lekarzem rodziny McFungus.

Ty, jako następca Sherlocka Holmesa, masz za zadanie zdemaskować przebiegłego zabójcę. Nie bądź jednak beztroski, zdarzy się jeszcze wiele morderstw, których można uniknąć.

Oto kilka wskazówek dla początkujących detektywów:
— naprzeciwko sejfu znajduje się sekretne przejście,
— w żakiecie Bentley'a znajdziesz klucz do pracowni profesora,
— staraj się przeszkodzić spotkaniu kucharki i Bentley'a,
— nie zwracaj uwagi na zrzędzenie majora,
— dowody rzeczowe umieszczaj w kopertach,
— nie pij płynu z małej butelki,
— nie otwieraj bomby, którą znajdziesz w łóżku,
— dokładnie przeszukaj sprzęty,
— nie noś ze sobą zbyt wielu przedmiotów naraz.

Firma: A.P. Software
Komputer: Commodore 64/128

(łc)

S.O.S.

Proszę o pomoc w grach: AZTEK, FINAL LAGACY oraz COLOSSAL ADVENTURE. Szukam też opisu do gry DIMENSION X w wersji na Atari.
Michał Spychalski Al. Mickiewicza 19/50 Łódź

O co chodzi w grach na Spectrum: RED MOON, THE BOGGIT, PRINCE OF MAGIC, KOBAYASHI NARU. Wymienię opisy, mapy i informacje.
Artur Lewandowski ul. Kaliska 9/3 02—316 Warszawa

Nie wiem, jak bawić się grą HYDRAULIK na Spectrum. Liczę na Waszą pomoc.
Dariusz Iwański ul. Opolska 42 47—312 Rogów Opolski

THE SECRET DIARY OF ADRIAN MOLE

Tego jeszcze nie było. Ani tak świetnej „komputerowej" filmu, ani tak dobrego pomysłu. Czyż można wymyślec coś wspanialszego? Lubię tekstówki, ale jeszcze nigdy nie przesiedziałem nad jedną tyle, co nad tą. Polecam ją każdemu, kto choć trochę zna angielski i dysponuje co najmniej dwudziestoma czterema wolnymi godzinami.

Jest to gra tekstowa bardzo różna od przyjętego standardu. Jej tekstowość polega przede wszystkim na bardzo rozwiniętej fabule i ogromnej ilości informacji w postaci tekstu. Fabuła przekazana jest w formie dziennika czternastoletniego chłopca Adriana Mole. Dziennik rozpoczyna się pierwszego stycznia któregoś roku i jest prowadzony do 31 grudnia. Urodziny Adriana przypadają 2 kwietnia. Gra podzielona jest na cztery części po trzy miesiące, każda z części zajmuje całą pamięć i jest wgrywana po skończeniu poprzedniej. W odróżnieniu od innych gier, rola gracza polega na wciskaniu któregoś z SHIFT-ów w celu przewijania tekstu. Jest on ilustrowany wieloma różnorodnymi rysunkami.

Dlaczego więc pozostaje grą? Otóż w pewnym momencie akcji narrator (Adrian) zatrzymuje się pytając, co ma zrobić lub którą z czynności powinien wybrać. Do wyboru są trzy warianty, uruchamiane wciśnięciem klawisza 1, 2 lub 3. Po wciśnięciu klawisza 4 można wydać jedną z komend LOAD SAVE DEMO PRINTER ON/OFF PICTURES ON/OFF lub HELP. Można też podać nazwisko lub przezwisko jednej z osób, by otrzymać wyczerpujące informacje na jej temat. Podanie którejś z cyfr powoduje powrót do gry.

Akcja toczy się w dwóch angielskich miastach. Przeplata się bardzo wiele wątków. Właściwym celem gry jest osiągnięcie 100% podobieństwa do oryginalnego Adriana. Aktualny wynik podawany jest co kilka dni.

Tekst napisany jest prostym językiem angielskim, zrozumiałym prawie dla każdego. Nie jest to język kwiecisty, oddaje jednak wiernie wszystkie sytuacje i humorystyczne zajścia. Prawie nie ma idiomów ani zwrotów specyficznych języka, dziennik czyta się lekko i z przyjemnością.

Nie mogę powiedzieć, jak kończy się pamiętnik, by nie psuć zabawy. Podam tylko garść najpotrzebniejszych informacji o głównych postaciach.

Nigel jest najlepszym przyjacielem Adriana. Ma jednak pstro w głowie, nie warto chodzić do niego na prywatki.

Pandora to nowa uczennica trzeciej D, pali papierosy. Na geografii siedzi z Adrianem, który się w niej podkochuje. Jej ojciec nie jest mleczarzem.

Ojciec Adriana jest chorowitym mężczyzną, lubi narzekać i leżeć w łóżku jedząc witaminę C.

Matka Adriana wykorzystuje syna jak może, głównie w pracach domowych. W pewnym momencie przeprowadza się do Sheffield z panem Lucasem.

Lucas uwiódł ją, lubi dawać Adrianowi prezenty, dobrze gotuje.

Scrunton jest dyrektorem szkoły Adriana, namiętnym palaczem. Nie lubi spóźnialskich.

Dock uczy angielskiego, ma stary samochód.

Kent jest szkolnym kolegą Adriana, często się spóźnia i bije kolegów, pali papierosy Benson & Hedges. Lepiej na niego nie skarżyć.

Baxter jest niedołężnym staruszkiem, którym Adrian musi się opiekować z ramienia Klubu Samarytanów. Chętnie pożycza pieniądze.

Brat Ludowic to przedstawiciel sekty Purple People. Lepiej z nim nie rozmawiać, za wszystko żąda zapłaty.

Babcia żyje w innym mieście, zawsze służy radą i pomocą.

Sproxton to nauczycielka angielskiego, często skarży dyrektorowi.

Ciotka Susan siedzi w więzieniu Holloway, pali cygara Panama.

Grace Pool jest znajomą ciotki Susan, lubi otrzymywać listy.

Gray to doktor Adriana, nie wierzy w jego dolegliwości.

I garść informacji o Adrianie: nie lubi sportu, uwielbia pisać i czytać wiersze, kocha się w Pandorze, lubi czekoladki Mars i motocykle, uważa się za niedocenianego intelektualistę, nie znosi palaczy, dentystów i egzaminów, jest uczulony na plastik.

Należy jeszcze nadmienić, że film pod tym samym tytułem jest tak popularny, że zakupiło go 70 krajów. Czy gra powtórzy jego sukces?

Firma: Level 9 & Mosaic
Komputer: ZX Spectrum 48/+

(mp)

Kochany Bajtku!

Bardzo lubię Wasze czasopismo i zawsze czekam niecierpliwie na kolejny numer. Z dużym zainteresowaniem śledzę również Listę Przebojów Gier Komputerowych. Do moich ulubionych gier należą: Panama Joe, Boulder Dash i Miss Pacman, a moim marzeniem jest posiadanie gry Barbarian na ZX Spectrum.

Chciałabym również opisać krótko na czym polega gra Miss Pacman. Posiada ona bardzo proste zasady i jest stosunkowo łatwa, dlatego gra się w nią długo i chętnie, po prostu nie denerwuje. Nasz bohater przypomina sympatyczną piłeczkę-kuleczkę, która wędruje przez labirynty zjadając wszystko co napotka po drodze. Kiedy nasza kuleczka zje wszystko co znajduje się w labiryncie wtedy „przechodzimy" do następnego etapu. W zamierzonym celu przeszkadzają nam „duszki-potworki", które czyhają na nasze życie. Na pewien czas możemy sobie obłaskawić duszki, jeżeli zjemy jedną z czterech dużych kulek, które znajdują się w rogach labiryntu. Od czasu do czasu w labiryncie pojawiają się owoce, które warto zjeść, otrzymujemy wtedy dodatkowe punkty (za zjedzenie banana nawet 5000 punktów), wiadomo, owoce to zdrowie!

W trakcie gry, poznajemy również małą historię naszego „człowieczka", jak poznał swoją narzeczoną i założył rodzinę. Akt 1: poznanie narzeczonej. Akt 2: wesołe zabawy. Akt 3: założenie rodziny (bocian przynosi małe dziecko!). I na tym w zasadzie kończy się gra. Aha, za 10.000 punktów otrzymujemy dodatkowe życie. Należy jeszcze życzyć grającemu, w tę grę, aby zawsze „doszedł" do 3 aktu, kiedy to wszystko kończy się dobrze.

Teresa Gajda
Zabełków, ul. Długa 13 47-460 Chałupki
lat 11, klasa IV
posiadany komputer ZX Spectrum

KRÓL I KRÓLOWA GIER



Monika Mieleszkiewicz lat 13
Szkoła Podstawowa nr 6 w Sopocie
Posiadany komputer: Timex
Ulubiona gra: Eskimo Ellie
Hobby: tenis, rower, narty, nauka angielskiego



Szymon Staszko, lat 15
od września Liceum Ogólnokształcące im. N. Żmichowskiej w Warszawie
posiadany komputer: Commodore 128
Ulubiona gra: Defender of the Crown
Hobby: muzyka, w szczególności gra na perkusji. Chciałby doskonalić swe umiejętności perkusisty przy pomocy swojego komputera i programu Microrhytm

---

Bardzo proszę o informacje, jak w grze TRAP DOOR zdobyć i usmażyć jajka oraz jak ugotować błotniaczki. W zamian służę opisami do gier PYJAMARAMA, COBRA, STALLONE i in.
Dawid Michałowski ul. Zatorska 35 51—215 Wrocław — Zakrzów

... nieśmiertelność w grze DRUID? Nie wiem ... grę JACKAL i przejść do czwartego po... w grze SIGMA 7. Wszystkie na C64
Manusz Grosiak Wężyki 96—514 Rybno woj. skierniewickie

... pol z grami GHOSTBUSTERS I WARHAWK NA ATARI 800 XL. Proszę o przysłanie mi opisów do nich.
Grzegorz Podsędek Ul Malczewskiego 19/54 93—154 Łódź

Nie mogę poradzić sobie z grą „GHOSTBUSTERS i bardzo proszę o pomoc.
Marek Schaffardzik Hlucińska 831; Sudgecovice 80; Okres Orawa: CSSR, PSC, 747 14.

Mam Timex 2048. Proszę o opisy do gier SILENT SERVICE, ELITE, V—VISITORS, INDIANA JOE, STAR FOX. Poszukuję też gier: EXOLON, SABOTEUR II, WAR, DAN DARE. W zamian oferuję wiele gier **Maciej Kuszkik ul. Kazimierza Wlk. 37/57 92—700 Bochnia**

Może ktoś przyśle mi opisy do gier: BIGGLES, TERMINUS, MONTY TAP, TOP GUN w wersji na Amstrad-Schneider 464
Robert Orliński ul. Marzanny 10 A 30—828 Kraków

Proszę o pomoc (dokładny opis) w grze PLANETARY DEFENSE. Odwdzięczę się instrukcjami do wielu innych gier.
Robert Domżalski ul. Stalowa 14 84—230 Rumia

Pilnie potrzebuję gry RAMBO i opis do niej. W zamian służę opisami do MONTEZUMA'S REV, ZORRO, PITFALL II. Jestem niepełnosprawny — trudno mi samemu coś zdobyć. Mam Atari 65XE
Sławomir Wuj ul. Krucza 82 A/10 40—756 Katowice 19

MINI-PAKIET GRAFICZNY NA KOM

Celem artykułu jest dostarczenie użytkownikowi Amstrada PCW narzędzi pozwalających na wykorzystanie możliwości graficznych sprzętu przy pisaniu własnych programów w języku Turbo-Pascal.

W pierwszej części artykułu przedstawiono następujące zagadnienia

1. realizacja, przy pomocy RSX'a, związanej ze sprzętem procedury PLOT — zapalenie i gaszenie dowolnego punktu ekranu,
2. współpraca RSX'ów z Turbo-Pascalem.

Uzupełnieniem tego materiału jest, umieszczony poza Klanem Amstrada, artykuł zawierający omówienie ogólnych algorytmów generacji krzywych na płaszczyźnie.

Druga część, która ukaże się za miesiąc, ma charakter bardziej związany ze sprzętem a konkretniej, z zastosowanym w 8-bitowych AMSTRADACH, mikroprocesorem Z80. Omówione zostaną w niej następujące tematy:

1. assemblerowa realizacja procedur kreślących proste i okręgi.
2. RSX umożliwiający graficzny zrzut ekranu na drukarkę
3. sposob dołączenia omawianych rozszerzeń do pierwszej wersji minipakietu.

PLOT

Wprowadzenie na różnych komputerach tego samego systemu operacyjnego CP/M pozwoliło na korzystanie na nich z tych samych programów. Niestety, z powodu różnorodności w realizacji funkcji graficznych na każdym praktycznie sprzęcie, poważnym problemem okazało się przenoszenie programów graficznych. Z tego powodu większość translatorów pracujących w CP/M nie ma żadnej grafiki. Dotyczy to np. Basic'a Mallarda czy Turbo-Pascala. Pewnym rozwiązaniem tego problemu jest standard graficzny GSX'a (ang. Graphics System eXtension). Niestety pakiet ten, dostępny także na AMSTRADzie, zajmuje dużo pamięci, zostawiając niewiele miejsca na program użytkownika. Celowe jest niekiedy stworzenie własnego systemu graficznego, na pewno o mniejszych możliwościach, ale za to mniej pamięciochłonnego. Ze względu na bliski związek ze sprzętem, konieczne jest zastosowanie kodu maszynowego. Turbo-Pascal posiada instrukcję INLINE, która pozwala na dołączenie fragmentów w kodzie do tekstu programu. Jest to rozwiązanie bardzo zwarte w zapisie i często bardzo wygodne. Jednak w przypadku Amstrada PCW, z powodu jego organizacji pamięci, wygodniej jest skorzystać z assemblera tworząc tzw. RSX'a (ang. Resident System eXtension), czyli rezydentne rozszerzenie systemu operacyjnego. Taka konstrukcja oprogramowania ma tę dodatkową zaletę, że ze stworzonego w ten sposób RSX'a można skorzystać w innych językach np. w Basic'u, Fortranie, itp.

Na listingu 1 przedstawiono zbiór PLOT.MAC realizujący, w formie RSX'a, dostęp do dowolnego bitu pamięci ekranu. Program ten wzorowany jest na RSX'ie przedstawionym w artykule „Komenda Plot w Basicu" („Komputer" 6/87). Nowa wersja jest rozszerzeniem poprzedniej o możliwość dołączenia assemblerowych procedur kreślących proste i okręgi (DRAW i CIRCLE). Zmieniono także sposób przekazywania parametrów, a całość zapisano w mnemonikach assemblera Z80. Dodano dokładniejszą kontrolę współrzędnej X (zakres 0..719) i odwrócono kierunek osi Y (0 na dole, 255 na górze ekranu).

WSPÓŁPRACA RSX'ów Z TURBO-PASCALEM.

Opisany wcześniej zbiór PLOT.MAC potrzebny jest do generacji zbioru typu RSX. Najprościej można to wykonać przy pomocy krótkiego zbioru wsadowego o nazwie MAKERSX.SUB:

M80 =$1
LINK $1 [OP]
ERA $1.RSX
REN $1.RSX=$1.PRL

Komenda:

A>MAKERSX SUB PLOT <CR>

utworzy zbiór PLOT.RSX. Na dysku A. musi znajdować się assembler M80 i dwa zbiory systemowe: SUBMIT.COM, LINK.COM

Posługując się programem systemowym GENCOM.COM, otrzymanego RSX'a można wykorzystać na różne sposoby:

1 A>GENCOM TURBO.COM PLOT.RSX<CR>

tworzy kompilator Pascala pozwalający, w trakcie kompilacji do pamięci, uruchamiać programy korzystające z funkcji PLOT (doskonałe przy pisaniu programu).

2 A>GENCOM DEMO.COM PLOT RSX <CR>

dołącza RSX'a do zbioru DEMO.COM.

3 A>GENCOM PLOT [NULL]

generuje zbiór PLOT.COM, który musi być uruchomiony bezpośrednio przed programem korzystającym z PLOT'a.

Nie polecam nikomu operacji:

A>GENCOM BASIC.COM PLOT.RSX

wykonanej na Basicu Mallarda, bo okaże się, że zamiast rozszerzonej wersji translatora, będzie miał zubożoną. Interpreter ten ma dołączone przez producenta inne RSX'y, a powyższa komenda odłącza je.

Kolejny problem to kwestia komunikacji programu użytkownika z RSX'em. Turbo-Pascal, dzięki funkcjom BDOS (także procedura) i ADDR pozwala na bardzo eleganckie rozwiązanie.

W zbiorze GRAPH.SYS (patrz artykuł „Algorytmy generacji krzywych na płaszczyźnie) znajduje się deklaracja:

var GPB : array (.0..4.) of integer;

Na zmienną GPB podstawiane są parametry (w przypadku PLOT'a są to współrzędne punktu X,Y i zmienna Pflag). Wywołanie procedury:

BDOS (76 ADDR (GPB));

spowoduje wyświetlenie lub zgaszenie punktu na ekranie, zależnie od wartości zmiennej Pflag.

W tym miejscu warto wspomnieć, że podstawową częścią systemu operacyjnego CP/M jest właśnie BDOS (Basic Disk Operating System) i że wszystkie konieczne operacje, nie tylko dyskowe, BDOS realizuje w postaci wywołań o określonych numerach. Przykładowo:

BDOS (2,48).

spowoduje wyświetlenie na ekranie znaku '0' o kodzie ASCII 48. Nie wszystkie numery funkcji BDOS zostały wykorzystane przez twórców CP/M'u. Zostało sporo wolnego miejsca, właśnie na RSX'y, pozwalające na przechwycenie odwołania do systemu operacyjnego. W ten sposób, oprócz dołączenia nowych możliwości mamy także sposobność modyfikacji odwołań systemowych.

Jarosław Młodzki

```
; ***********************************************
; *            Zbiór PLOT.MAC                   *
; *              ver. 1.0                       *
; *     (C) Jarosław Młodzki  Luty 1988         *
; *                                             *
; * Procedura dostępu do pamięci ekranu         *
; * dla komputerow AMSTRAD PCW 8256/8512        *
; * Procedura zapala lub gasi punkt na          *
; * ekranie.                                    *
; *                                             *
; * parametry:                                  *
; *  rejestr c = 76                             *
; *  rejestr de - adres bloku parametrów        *
; *  przekazywanych z programu uzytkownika      *
; *       numer  nazwa   znaczenie              *
; *        1     Pflag   =0 zgaszenie pktu      *
; *                      =1 zapalenie pktu      *
; *       2,3    x,y     współrzędne pktu       *
; ***********************************************
     .z80
scrrun equ 0E9H
     cseg
; *********  nagłowek RSX'a
     ds    6
     jp    start
next: db   0C3H,0,0,0,0,0FFh,0
     db    'GRAPHICS',0,0,0
; *********
start:          ; sprawdzenie nr funkcji BDOS'a
     ld    a,c
     cp    76
     jp    z,plt    ; skocz do miejsca przesłania
;                       parametrów PLOT'a
     cp    75
     jp    z,drw    ; skocz do miejsca przesłania
;                     parametrów DRAW'a
```

PUTER AMSTRAD PCW 8256/8512

```
      cp    74
      jp    z,crc    ; skocz do miejsca przesłania
;                      parametów CIRCLE'a
      jp    next
; ************** przesłanie parametrow procedury
;                PLOT do wspólnej pamięci
plt:  call evadr
      ld    hl,PFlag
      ld    bc,6
      ex    de,hl
      ldir
      ld    bc,plot
      ld    hl,(y1)
      ld    de,(x1)
      call jphl
      dw    scrrun
      ret
; ************* przesłanie parametrów procedury
;               DRAW do wspólnej pamięci
;  ( assemblerowa realizacja procedury kreślącej
;    linię )
drw:  call evadr
      ld    hl,PFlag
      ld    bc,10
      ex    de,hl
      ldir
      ld    bc,draw
      call jphl
      dw    scrrun
      ret
; ************* przesłanie parametrów procedury
;               QCIRCLE do wspólnej pamięci
;  ( assemblerowa realizacja procedury kreślącej
;    okrag - odpowiednik QCIRCLE ,)
crc:  call evadr
      ld    hl,PFlag
      ld    bc,8
      ex    de,hl
      ldir
      ld    bc,circle
      call jphl
      dw    scrrun
      ret
; **************
jphl: db    0c3h
adr:  dw    0
; ************* pomocnicza procedura wyznaczenia
;        adresu funkcji BIOS o numerze 30 (USERF)
evadr:
      ld    hl,(0001)
      ld    bc,87
      add   hl,bc
      ld    (adr),hl
      ret
; **********************************************
; procedura PLOT  ( de=x, hl=y, pflag)
plot: ld    b,1
      ld    a,3
      and   d
      ld    d,a    ; x w zbiorze (0..1023)
      ld    hl,719
      xor   a
      sbc   hl,de
      ret   m      ; jeśli x>719 - nie ma plot'a.
      ld    a,255
      sub   b
      ld    l,a    ; y od dołu do góry
      ld    h,0    ; y w przedziale (0..255)
      add   hl,hl
      ld    bc,0b600h
      add   hl,bc
      ld    c,(hl)
      inc   hl
      ld    b,(hl)
      ld    a,c
      and   0f8h
      ld    l,a
      ld    h,b
      add   hl,hl
      add   hl,de
      ld    a,l
      and   0f8H
      ld    l,a
      ld    a,c
      and   7
      or    1
      ld    l,a
      ld    a,e
      and   7
      inc   a
      ld    b,a
      xor   a
      scf
lxxp: rra
      djnz  lxxp
      ld    c,a
      ld    a,(PFlag)
      cp    1
      jp    nz,reset
      ld    a,c          ; zapalenie pktu
      or    (hl)
      ld    (hl),a
      ret
reset:                   ; zgaszenie pktu
      ld    a,c
      cpl
      and   (hl)
      ld    (hl),a
      ret
; ************** obszar zmiennych roboczych
PFlag:ds    2
x1:   ds    2
y1:   ds    2
x2:   ds    2
y2:   ds    2
fx:   ds    2
fy:   ds    2
dx:   ds    2
dy:   ds    2
va:   ds    2
vx:   ds    2
vy:   ds    2

; ***********************************************
; procedura DRAW kreśląca linię z pktu
; x1,y1 do pktu x2,y2
draw:
; miejsce na assemblerowa wersje  procedury DRAW
      ret

; ***********************************************

; ***********************************************
; procedura CIRCLE kreśląca okrąg (x0,y0,R)
circle:
   miejsce na assemblerową wersję procedury CIRCLE
      ret

; ***********************************************
      end
```

(część druga, w której wszystko dobrze się kończy)

CZY KOMPUTER UMIE LICZYĆ?



Miesiąc temu przekonaliśmy się, że jeśli nie wymagamy cudów, nie próbujemy wykonywać obliczeń, których wynik jest za duży i nie daje się poprawnie zapisać w pamięci maszyny, to nie ma podstaw do kwestionowania arytmetycznych zdolności komputera. Przynajmniej w części, bo jeszcze nie wszystkie problemy udało się rozwiązać.

Jeden z problemów, które powstały, to zapisanie w pamięci bardzo dużych liczb — ilość potrzebnego miejsca wzrasta z wielkością liczby

Szukanie rozwiązania chciałbym znowu zacząć od przypomnienia metody tradycyjnej, używanej przy zapisie liczb na papierze. Przy bardzo dużych (lub bardzo małych) liczbach, aby uniknąć pisania długich ciągów zer, używamy tzw. czynnika skalującego. Zamiast 1521000000 piszemy $1521 \cdot 10^{6}$ Zauważmy, ze można to zapisać również jako: $1.521 \cdot 10^{9}$*) lub jako $0.1521 \cdot 10^{10}$. Inaczej mówiąc, czynnik skalujący wskazuje przez jaką potęgę dziesięciu trzeba pomnożyć to co przed nim stoi, aby otrzymać pełną postać liczby (jej tradycyjny zapis). Moze jeszcze kilka przykładów:

$0.00000182 = 1.82 \cdot 10^{-6} = 0.182 \cdot 10^{-5}$
$4823198.221 = 0.4823198221 \cdot 10^{7}$
$-119750000000 = -0.11975 \cdot 10^{12}$
$10000000 = 0.1 \cdot 10^{8}$
$1 = 0.1 \cdot 10^{1}$
$0.1 = 0.1 \cdot 10^{0}$ (pamiętamy, że $10^{0}=1$)

Ostatnie dwa przykłady mają nam uświadomić, że podany sposób pozwala zapisywać wszystkie liczby, nie tylko bardzo duże lub małe.

Wprowadźmy następujące nazwy: wykładnik potęgi to cecha, zaś część mnożona przez potęgę, to mantysa.

$$2.168943 \cdot 10^{11}$$

↗ mantysa ↘ cecha

Jeśli już umówimy się, że stosujemy tę postać zapisu (zwaną także postacią wykładniczą), to nie musimy pisać "$\cdot 10$", wystarczy podać dwie liczby, np. 0.2168943;11. Wtajemniczeni natychmiast rozumieją, że chodzi o $0.2168943 \cdot 10^{11}$. Możemy nawet być jeszcze bardziej oszczędni. Umawiamy się dodatkowo, że cecha (wykładnik), będzie zawsze wybrana tak, aby mantysa była ułamkiem z pierwszą cyfrą po kropce niezerową. Wtedy możemy podać tylko 2168943;11. Oznacza to dalej to samo: $0.2168943 \cdot 10^{11}$, gdyż potrafimy się domyśleć gdzie postawić zero i kropkę.

Co możemy uzyskać dzięki takiemu zapisowi? Wszystkie liczby i bardzo duże i całkiem malutkie zapisujemy na takim samym kawałku papieru, np. raz na zawsze przeznaczmy dziesięć kratek na mantysę, 3 kratki na cechę. W sumie 13 kratek, a pozwoli to zapisać nawet:
346000000000000000000000000000,
bo mamy np. $0.346 \cdot 10^{28}$, czyli nasza mantysa to 0.346, zaś cecha 28. Okazuje się, że nawet nie przybliżyliśmy się do granicy dopuszczalnych wielkości (granicę tę określa z grubsza liczba składająca się z jedynki i tysiąca zer!).

Czym musimy zapłacić za takie możliwości. Niestety stratą dokładności. Liczba składająca się z kilkunastu cyfr wcale nie musi mieć na końcu zer. Np. 3261524232652 zostanie zapisane u nas jako: 0.3261524232;13 (czyli $0.3261524232 \cdot 10^{13}$). Widać, że trzy ostatnie cyfry (652) nie zmieściły się (mamy tylko 10 kratek) i zostały stracone. Aby się o tym upewnić wykonajmy z powrotem przejście z naszej reprezentacji na pełną postać: $0.3261524232 \cdot 10^{13} = 3261524232000$. Tym razem jednak nie wyciągnę z rękawa żadnego cudownego sposobu na ominięcie tej niedogodności. Tak musi być, pokazana wyżej strata jest nieunikniona, jeśli chcemy na ograniczonej przestrzeni zapisywać bardzo duże liczby.

Można by znowu próbować przeznaczyć więcej miejsca na mantysę, ale to niestety nic nie da. Spróbujmy podzielić na papierze np. 1 przez 3. Otrzymujemy: 0.3333333..... Dzielenie to nigdy się nie kończy, zawsze otrzymujemy wynik 3 i resztę 1, która po podzieleniu daje 3 i 1, itd. Po prostu ułamek $^{1}/_{3}$ nie da się dokładnie zapisać na ustalonej liczbie pozycji. Jego reprezentacja dziesiętna (i oczywiście także dwójkowa, czy w układzie o jeszcze innej podstawie) jest nieskończona, więc siłą rzeczy każda kartka będzie za wąska na to, aby tę reprezentację zapisać dokładnie. Nie możemy mieć również złudzeń, że $^{1}/_{3}$ jest jedyną tak złośliwą liczbą. Liczb o takich, nigdy nie kończących się zapisach jest ogromnie dużo (nawet nieskończenie wiele) i zawsze będą one występowały w naszych rachunkach.

Tak więc ze stratą dokładności musimy się pogodzić i już, o tym że nie jest to strata bardzo bolesna powiemy sobie za chwilę. Teraz musimy nauczyć się wykonać działania arytmetyczne na liczbach zapisanych w postaci wykładniczej. Dla ułatwienia weźmy skromniutką reprezentację: mantysa 5 cyfr po kropce (inaczej mówiąc 5 cyfr znaczących), cecha 2 cyfry. Żeby zwiększyć czytelność będziemy zapisywać czynnik skalujący w pełnej postaci. Dodajmy $0.58321 \cdot 10^{7}$ i $0.83181 \cdot 10^{7}$.

$0.58321 \cdot 10^{7}$
$0.83181 \cdot 10^{7}$

Wystąpiło to samo co miesiąc temu — przeniesienie na najbardziej znaczącej pozycji. Pamiętamy sprzed miesiąca, że w takiej sytuacji, przy klasycznym zapisie pozycyjnym, wynik jest zupełnie bezwartościowy (tracimy najbardziej znaczącą cyfrę!) i komputer przerywa obliczenia z sygnalizacją błędu. Popatrzmy co stanie się w zapisie wykładniczym. Zapiszmy wynik w tej samej postaci co argumenty (zero przed kropką): $0.141502 \cdot 10^{8}$. Ponieważ jednak nasza kartka na zapisywanie mantysy ma miejsce tylko na 5 cyfr po kropce musimy to zapisać tak: $0.14150 \cdot 10^{8}$. Też straciliśmy jedną cyfrę (ostatnia 2 zniknęła), ale tym razem jest to cyfra najmniej znacząca, nasz wynik ma sens. Przeniesienie w dodawaniu spowodowało tylko zwiększenie cechy i nieznaczną stratę dokładności.

Popatrzmy raz jeszcze. Zrobiliśmy rzecz następującą: gdy otrzymaliśmy mantysę 1.41502, to przesunęliśmy kropkę dziesiętną w lewo, tak aby dostać 0.141502. Własnie to przesunięcie pozwoliło nam ocalić w miarę poprawny wynik. Stąd pochodzi nazwa takiej arytmetyki — zmiennopozycyjna lub zmiennoprzecinkowa (ang. floating-point)

Przejdźmy teraz do komputerów, aby przekonać się, że nie na próżno męczyliśmy się z tym nienaturalnym dla człowieka zapisem. Arytmetyka zmiennoprzecinkowa jest podstawowym sposobem wykonywania w komputerach obliczeń na liczbach rzeczywistych. Oczywiście mantysa i cecha zapisane są dwójkowo, cecha określa potęgę dwójki, przez którą należy pomnożyć mantysę. Jednak poza tymi różnicami wszystko przebiega dokładnie tak samo jak przed chwilą opisaliśmy dla systemu dziesiętnego.

Tak jak każda kartka, również komputer ma skończone wymiary i skończoną pojemność. W takim razie zawsze przy wykonywaniu obliczeń należy liczyć się ze stratą dokładności. Mówimy, że pod-

czas obliczeń występują błędy zaokrągleń.

Nie jest to problem występujący tylko w teorii. Sami spotykaliśmy go w praktyce, przy porównywaniu różnych programów obliczających wartość wielomianu. Ponieważ każdy program obliczał wartość za pomocą innego zestawu operacji, to i błędy zaokrągleń były inne w każdym z nich. Stąd właśnie niewielka różnica pomiędzy wynikami. Dodajmy od razu: różnica na jedenastej cyfrze znaczącej, więc chyba bez większego znaczenia w praktyce.

Zapamiętajmy teraz bardzo ważną rzecz, to co powiedzieliśmy o występowaniu błędów zaokrągleń dotyczy absolutnie wszystkich komputerów. Tych, które być może macie przed sobą na biurku i tych wielkich maszyn, używanych już od wielu lat, także do bardzo ważnych obliczeń, od wyników których zależy bardzo wiele. Wszystkich bez wyjątku elektronicznych maszyn cyfrowych, bo wszystkie działają na tej samej zasadzie i tak samo reprezentują liczby.

Jak w takim razie jest możliwe, że komputery, tak niedokładne, od wielu lat są z powodzeniem stosowane w praktyce, mało tego, stały się w mowie potocznej symbolem dokładności.

Pierwsza odpowiedź jest ilościowa: długość pola na zapis danych w komputerze, choć ograniczona, jest zwykle tak duża, że pozwala uzyskać w razie potrzeby do kilkunastu dokładnych cyfr znaczących. Ludzie liczący ręcznie czy na suwaku logarytmicznym zwykle nie mieli możliwości lub cierpliwości aby uzyskać taką dokładność, szczególnie jeśli działań do wykonania było wiele tysięcy. Przed komputerami też wykonywano obliczenia, robili to np. wyspecjalizowani rachmistrze. Oni też musieli zaokrąglać. Komputer, dzięki swej szybkości jest w stanie zaokrąglać na dalszych pozycjach po przecinku, czyli uzyskiwać więcej cyfr znaczących, dlatego wprowadzenie komputerów stanowiło ogromny postęp.

Nie jest to jednak wyjaśnienie w pełni wystarczające. Choćby dlatego, że dzięki dużej szybkości komputer może wykonywać dużo dłuższe procesy obliczeniowe, a więc może wystąpić bardzo wyraźnie tzw. akumulacja (odkładanie się) błędu **).

Dlatego wszystkie metody obliczeniowe stosowane na komputerach są bardzo skrupulatnie badane przez naukowców. Zajmuje się tym analiza numeryczna — wielka gałąź nauki z pogranicza czystej matematyki i informatyki. Dzięki jej wynikom mamy w tej chwili bardzo dobre rozeznanie, pozwalające do konkretnych zadań dobierać metody gwarantujące otrzymanie, mimo występowania błędów zaokrągleń, wyników odpowiednio dokładnych.

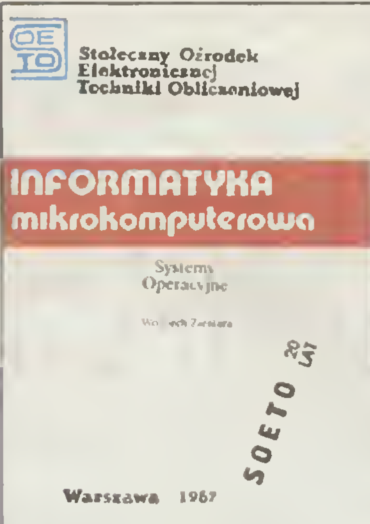
Ostatecznie więc, możemy chyba stwierdzić, że komputer w dobrych rękach potrafi liczyć bardzo dobrze. A jeśli trafi w ręce ignoranta? Cóż, jeśli ktoś przytłucze sobie paznokieć przy wbijaniu gwoździ, nie powinien chyba mieć pretensji do młotka. Wydaje mi się, że tak samo należy podchodzić do używania komputerów.

Miesiąc temu i dziś zajmowaliśmy się raczej teoretyczną stroną zapisu liczb i operacji na nich. Za miesiąc postaramy się mocniej powiązać nasz dorobek z praktyką.

***) Zgodnie z panującym w informatyce zwyczajem część ułamkowa od części całkowitej liczby dziesiętnej oddzielam kropką, a nie przecinkiem.**
****) Jeśli wynik jednej operacji jest następnie argumentem następnej, to błąd może narastać. Przy wykonywaniu wielu operacji zjawisko narastania błędu staje się bardzo niebezpieczne.**

Andrzej Pilaszek

PRZECZYTALIŚMY TO DLA WAS



Nasza redakcja codziennie otrzymuje listy i telefony, w których użytkownicy Atari dzielą się z nami swoimi spostrzeżeniami lub zwracają się z prośbą o rady. Największe zainteresowanie zwykle budzi wykorzystanie funkcji PEEK i POKE, dzięki którym można czasami zaskakująco wpłynąć na przebieg programu. Od eksperymentowania z nimi zaczyna się prawdziwa przygoda.

Pewne informacje o systemie kontrolującym pracę Atari XL/XE oraz ogólna mapa pamięci znajdują się w książce W. Miguta „Atari BASIC". Pozwalają one zorientować się w funkcjonowaniu komputera, nie wystarczają jednak, gdy dociekliwy hobbysta zacznie zadawać sobie bardziej szczegółowe pytania.

Stąd właśnie zrodził się pomysł napisania książki wnikliwie analizującej pracę systemu operacyjnego Atari XL/XE. Ukazująca się właśnie nowa publikacja SOETO, „Mapa pamięci Atari XL/XE" Wojciecha Zientary, jest pierwsza z czterech książek opisujących oprogramowanie systemowe tego komputera.

Zaciekawi ona wszystkich POKE-rzystów, którzy teraz na pewno nie będą musieli na ślepo poszukiwać adresów i odgadywać znaczenia ważniejszych komórek pamięci.

Książka nie tylko ułatwia wymyślanie sztuczek do programu. Przeznaczona jest przede wszystkim dla tych, którzy zamierzają gruntownie poznać funkcjonowanie Atari i wykorzystać tę wiedzę do pisania dobrego i ciekawego oprogramowania. Umożliwia natychmiastowe zastosowanie gotowych procedur z pamięci komputera, których samodzielne poszukiwanie zajęłoby mnóstwo czasu.

Pierwsza część „Mapy pamięci Atari XL/XE" zawiera opis procedur inicjujących pracę systemu oraz przerwań maskowalnych i niemaskowalnych. W drugiej znajdziemy informacje o procedurach wejścia i wyjścia obsługi urządzeń zewnętrznych (w tym klawiatury, ekranu, drukarki, magnetofonu i stacji dysków). Dowiemy się z niej także, jak zarządzana jest grafika graczy i pocisków oraz jak tworzony jest dźwięk. Trzeci tom przeznaczono dla użytkowników stacji dysków. Omawiane są tu różne dyskowe systemy operacyjne (DOS 2.05, DOS 2.5 i DOS X...) z uwzględnieniem procedur inicjujących i procedur wejścia i wyjścia. Ostatnia część będzie bardzo pomocna wszystkim posługującym się BASIK-iem. Dokładne poznanie działania intepretera, sposobu realizacji funkcji i gospodarowania pamięcią daje ogromne korzyści. Znajomość BASIC-a od podszewki jest wręcz niezbędna, gdy zależy nam na szybkości działania programu lub oszczędności pamięci.

„Mapa pamięci Atari XL/XE" jest książką, która pogłębia wiedzę użytkownika Atari. Szczególnie cenna będzie dla osób poważnie myślących o samodzielnym pogramowaniu. Podobne materiały znajdują się zawsze pod ręką w firmach piszących oprogramowanie. Producenci sprzętu dbają o zagwarantowanie łatwej ich dostępności, publikując je często przed ukazaniem się na rynku nowego modelu komputera.

Procedury zamieszczone w książce mają format przyjęty w asemblerze MAC/65. Niezbędne jest wcześniejsze poznanie i zrozumienie zasad programowania w tym języku. Do nauki asemblera polecałbym książkę Jana Ruszczyca „Asembler 6502" wydaną również przez SOETO.

(j.j.)

Wojciech Zientara, „Mapa pamięci Atari XL/XE", „Podstawowe procedury systemu operacyjnego", SOETO, Warszawa, ul. Hoża 50, 1988, wyd. I, nakład 5000 egz., cena 920,- zł.

Wojciech Zientara, „Mapa pamięci Atari XL/XE". Procedury wejścia/wyjścia", SOETO, Warszawa, ul. Hoża 50, 1988, wyd. I, nakład 5000 egz., cena 1100,- zł.

Wojciech Zientara, „Mapa pamięci Atari XL/XE. Dyskowe systemy operacyjne", SOETO, Warszawa, ul. Hoża 50, 1988, wyd. I, nakład 5000 egz., cena ok. 1000,- zł.

Wojciech Zientara, „Mapa pamięci Atari XL/XE". „Procedury interpretera Basica", SOETO, Warszawa, ul. Hoża 50, 1988, wyd. I, nakład 5000 egz., cena ok. 1000,- zł.

PROGRAMY KOMPUTEROWE KRAJOWEJ AGENCJI WYDAWNICZEJ

Od grudnia 1986 r. Krajowa Agencja Wydawnicza w Warszawie produkuje programy komputerowe do masowego użytku. Wszystkie programy wydawane są w wersjach kasetowych na komputery ZX SPECTRUM, ATARI serii XL i XE, Commodore 64. Dotychczas nakładem Agencji ukazały się następujące pozycje.

- BIORYTMY/NUMERY SZCZĘŚCIA — programy rozrywkowe
- NIM II/TIXO — mało znane gry logiczne
- KÓŁKO I KRZYŻYK
- HEXAN — gra labiryntowa połączona z konkursem
- FUNKCJA KWADRATOWA — program edukacyjny
- RACHUNEK PRAWDOPODOBIEŃSTWA cz. I — program edukacyjny
- BASIC TEST — program edukacyjny dla początkujących programistów
- ATARI BASIC — programy z książki ATARI BASIC wydanej również przez KAW (także w wersji dyskowej). Pozycje tylko dla komputera ATARI.

W najbliższym czasie spodziewane są na rynku nowe programy:

- CZARNY PONIEDZIAŁEK — gra giełdowa dla wszystkich. Symulowane są tu transakcje zawierane na giełdzie. Uczestnicy gry mają za zadanie uzyskać maksymalny zysk z kupna i sprzedazy akcji znanych firm. Gra pełna jest zabawnych niespodzianek, zwłaszcza dla szczególnie wytrwałych graczy. Gra posiada również pewne walory dydaktyczne.
- MŁYNEK — jedna z najszybszych gier zręcznościowych, zrealizowanych kiedykolwiek na komputerze. Żadnemu z zaproszonych do testowania specjalistów nie udało się przejść wszystkich jej etapów. Ta gra stać się może prawdziwym przebojem.
- HANDEL ZAGRANICZNY — gra operacyjna. Grający stają się dyrektorami przedsiębiorstw handlowych, kierującymi przy pomocy komputera działalnością swoich firm. Mimo znacznego stopnia komplikacji prowadzonych działań, gra pozostaje atrakcyjna dzięki m.in. doskonałej grafice, przemyślanej organizacji i zaskakującym, zabawnym zwrotom sytuacji. Można ją polecić wszystkim miłośnikom niebanalnej rozrywki w wieku od lat 11 do 100.
- RACHUNEK PRAWDOPODOBIEŃSTWA — cz. II — kontynuacja cyklu programów edukacyjnych, zdobywających rynek doskonałą koncepcją dydaktyczną, atrakcyjną grafiką i niezawodnością. Programy tej serii dostosowane są do wymagań programu nauczania matematyki w szkołach podstawowych i średnich.
- SŁOWNIK — program użytkowy dla osób korzystających z anglojęzycznych tekstów z zakresu elektroniki. Zawiera wiele potrzebnych słów i wyrażeń, pozwala również na wprowadzenie własnych. Wydatnie skraca czas tłumaczenia tekstów.
- FAC — niekonwencjonalna gra zręcznościowa, wymagająca również umiejętności logicznego myślenia. Zabawny stworek — Fac prowadzi nas przez wiele komnat tajemniczego labiryntu, po przejściu komnaty pojawia się następna i gra zdaje się nie mieć końca...

Programy Krajowej Agencji Wydawniczej zakupić można w Salonach Wydawniczych KAW oraz w sieci KMPiK zajmujących się sprzedażą programów komputerowych. Stałą sprzedaż kaset i dyskietek prowadzi również współpracująca z KAW i znana niemal wszystkim fanom informatyki Księgarnia „Elektronika" w Warszawie przy ul. Mokotowskiej 51/53.

(szar)

TRÓJWYMIAROWE

W ostatnim czasie w różnych czasopismach pojawiały się publikacje poświęcone trójwymiarowym wykresom funkcji. Początkowo ograniczały się one do samego wydruku, z dopisaną informacją do czego służy zamieszczony program, nie tłumacząc jak jest to realizowane. Ostatnio w MIKROKLANIE zamieszczono cykl artykułów na ten temat, ale z punktu widzenia programisty — amatora był on chyba zbyt trudny, a już na pewno nie dawał gotowej odpowiedzi na pytanie „jak". Mam zamiar odpowiedzieć na to pytanie wszystkim tym, którzy zechcą przeczytać i zrozumieć ten artykuł w całości.

Żeby móc uporać się z tym zagadnieniem potrzebne nam będą podstawowe wiadomości na temat funkcji trygonometrycznych, trochę wyobraźni przestrzennej i jakikolwiek komputer z grafiką wysokiej rozdzielczości, bo chociaż zamieszczony program był napisany na Spectrum, da się go uruchomić na większości używanych w Polsce mikrokomputerów.

Zacznijmy od najprostrzego zadania — narysujmy płaszczyznę (możemy się umówić że jest to kartka papieru). Żeby obraz był czytelniejszy, zaznaczmy na nim jeszcze trzy osie — x, y, z. Na takim rysunku nie widać niestety, czy gdzieś na środku nie ma jakiejś wypukłości — żeby się o tym przekonać dorysujmy jeszcze kilka poziomych odcinków. Mamy to wszystko na rysunku nr 1. To co zrobiliśmy, możemy opisać w następujący sposób — powierzchnię, którą chcemy narysować, pokroiliśmy w paski i nanieśliśmy na rysunek ślady przecięć. Gdyby ślady te były krzywymi, mielibyśmy obraz jakiejś powierzchni, innej niż płaszczyzna (obejrzyj zamieszczone wydruki).

Teraz zbliżamy się do pierwszego z dwóch najistotniejszych punktów całego artykułu. Musimy sobie postawić pytanie — skąd mamy wiedzieć, w którym miejscu należy zaznaczyć kolejny ślad przecięcia? Obliczanie jak ma on wyglądać jest proste, są to dokładnie takie same rachunki, jak te potrzebne do narysowania wykresu jednej zmiennej. Ale gdzie narysować ten wykres? Jak znaleźć miejsce, w którym ma się on zacząć — czyli, innymi słowy, jak znaleźć wartości, o które należy przesunąć każdy jego punkt względem miejsca, w którym narysowalibyśmy go gdyby nie był fragmentem wykresu trójwymiarowego? Spójrzmy na rysunek nr 2. Jeżeli przypomnimy sobie odrobinę matematyki, okaże się, że zadanie nasze polega na wykonaniu rzutu — musimy znaleźć sposób rzutowania punktów mających współrzędne (x, y, z) na płaszczyznę wyznaczoną przez osie x i z. Innymi słowy, musimy znaleźć sposób obliczenia na podstawie znanych wartości (x, y, z) wartości nowych współrzędnych — x_1 i y_1, odnoszących się do płaszczyzny, na której ma się znaleźć wykres (w naszym przypadku oś x_1 pokrywa się z osią x, a oś y_1 z osią z). Jak to zrobić? Rzecz w tym, że już to zrobiliśmy, i to wcale o tym nie wiedząc! Przecież zaznaczony przez nas intuicyjnie punkt P na rysunku nr 2 ma, oprócz współrzędnych (x, y, z), również jakieś współrzędne (x_1, y_1), trzeba się tylko zastanowić nad tym, jak je z tego rysunku odcyfrować. Żeby było łatwiej zrozumieć jak to się robi, spójrzmy na rysunek nr 3, na którym jest zaznaczone wyłącznie to co będzie nam potrzebne. Możemy na jego podstawie, korzystając z definicji sinusa i cosinusa napisać następujące wzory:

$$x1 = x + y \cdot \cos\alpha$$

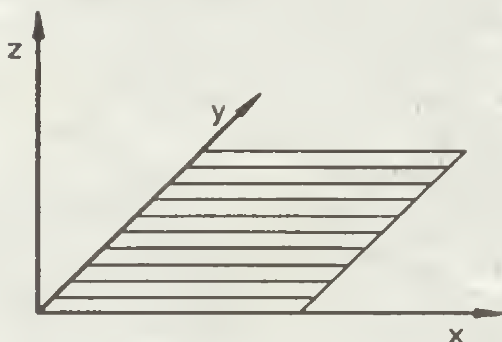
$$y1 = z + y \cdot \sin\alpha$$

Na podstawie tych dwóch zależności możemy teraz każdy punkt z przestrzeni x, y, z nanieść na płaszczyznę x_1, y_1, a co za tym idzie, jesteśmy gotowi do rysowania śladów przecięć w tych miejscach, w których powinny się one znaleźć. Wprawdzie nie uwzględniliśmy perspektywy, która powinna „skrócić" odcinek o długości y, ale — jak widać na wydrukach — nie ma to wielkiego znaczenia, chociaż uważny obserwator może kręcić nosem, że kształt przedstawiony na wykresie jest nieco zdeformowany. Można to z łatwością poprawić, mnożąc wartość y przez odpowiednio dobrany współczynnik, równy około 0.7.

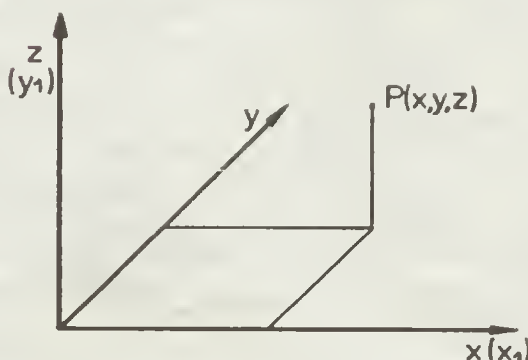
Umiemy już narysować wykres trójwymiarowy, lecz nie potrafimy jeszcze doprowadzić do tego, żeby te jego części, które powinny być zasłonięte, były rzeczywiście niewidoczne. Na ogół artykuły kończą się w tym miejscu, uwagą o tym, że jest to trudne lub bardzo trudne zadanie. My spróbujemy pójść dalej i zgłębić również ten problem.

Musimy znaleźć sposób na określenie tego, czy dany punkt ma być zaznaczony czy też nie. Pierwszą rzeczą, na którą warto zwrócić uwagę jest fakt, że ślad przecięcia najbliższy nas możemy narysować w całości, gdyż na pewno nie będzie przez nic zasłonięty. Ale co będzie przy rysowaniu następnych? Pomóżmy sobie rysunkiem 4. Są na nim dwie krzywe — a to ta najbliższa nas, b to następna, leżąca tuż za nią. Nietrudno zauważyć, a jest to druga rzecz istotna dla rozwiązania problemu, że zaznaczone mogą być wszystkie punkty krzywej b leżące poza zakreskowanym obszarem. Wynika z tego, że gdybyśmy potrafili dla każdego punktu leżącego na krzywej b sprawdzić, czy znajduje się on w zakreskowanym polu, czy poza nim (czyli czy znajduje się „w cieniu" rzucanym przez ustawioną pionowo płaszczyznę ograniczoną z góry śladem naszego cięcia a), mielibyśmy problem rozwiązany. A jak to zrobić? Popatrzmy jeszcze raz na rysunek. Punkty A i B leżą na krzywych a i b, przy czym mają tę samą współrzędną x_1, a różne y_1. Punkt A znajduje się nad punktem B, i ta właśnie informacja oznacza, że punkt B znajduje się w zakreskowanym polu, podobnie jak dla punktów A_1 i B_1, to, że punkt B_1 leży powyżej punktu A_1 oznacza, że B_1 nie należy do zakreskowanego obszaru. Innymi słowy — zanim zaznaczymy kolejny punkt, musimy sprawdzić, czy w rządku, w którym ma się on znajdować nie ma już powyżej jakiegoś punktu. Jeśli jest — nie zaznaczamy, jeśli nie ma — zaznaczamy. Ot i cała filozofia. Jeżeli będziemy rysować w ten sposób wszystkie kolejne krzywe leżące coraz dalej od nas, to uzyskamy pożądany efekt, czyli narysowany wykres trójwymiarowy z zasłoniętymi odpowiednimi fragmentami.

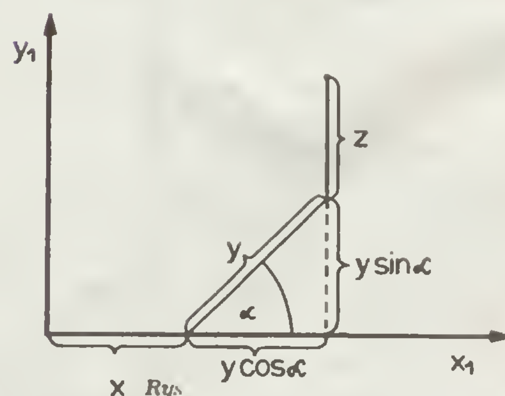
Sposobów realizacji tego pomysłu jest sporo, poczynając od najbardziej dosłownych, korzystających z



Rys. 1

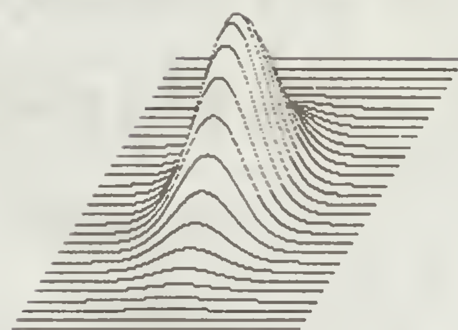


Rys. 2



Rys. 3

WYKRESY



funkcji POINT (x, y) lub jej odpowiednika, a kończąc na bardzo prostym rozwiązaniu, które zaproponuję.

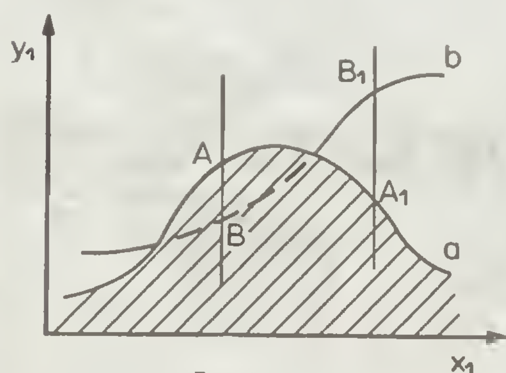
Oto program:

```
10 DIM M(255)
20 FOR Y=1 TO 141 STEP 5
30 FOR X=1 TO 141
40 LET Z=80*EXP(-1E-03*((Y-70)*(Y-70)+(X-70)*(X-70)))
50 LET X1=X+Y*COS(PI/4)
60 LET Y1=Z+Y*SIN(PI/4)
70 IF Y1>=M(X1) THEN LET M(X1)=Y1: PLOT X1,Y1
80 NEXT X
90 NEXT Y
```

Uwaga! Zanim wprowadzisz i uruchomisz ten program, przeczytaj artykuł do końca!

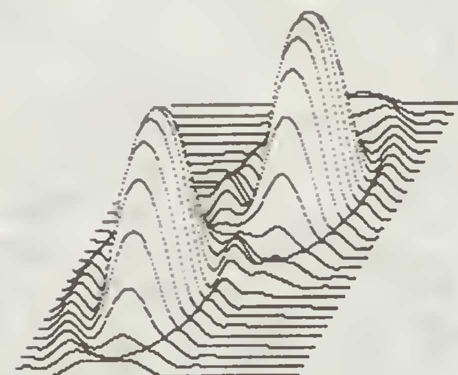
Zastosowaliśmy następujący algorytm — dla każdego pionowego rządka pikseli zapamiętujemy w tablicy M współrzędną pionową najwyżej zaznaczonego punktu. Współrzędne każdego kolejnego punktu są testowane — czy punkt ten miałby być zaznaczony powyżej dotychczas najwyższego w danym rządku, czy poniżej, i od wyniku testu uzależnione jest jego zaznaczenie i zapamiętanie położenia jako nowego najwyższego. Jest to realizowane w linii 70. W linii 40 obliczana jest wartość funkcji, której wykres rysujemy, w liniach 50 i 60 następuje przeliczenie współrzędnych (x, y, z) na (x_1). W liniach 20 i 30 otwieramy pętle, które „kroją" powierzchnię na paski. Jako wartość kąta α przyjąłem 45° — można oczywiście poeksperymentować z innymi wartościami, chociaż wydaje mi się, że biorąc pod uwagę rozdzielczość ekranu i wynikające z niej kłopoty z czytelnością obrazu, kąt 45° jest najlepszy (przynajmniej na Spectrum).

Każdy, kto chociaż raz zajmował się pisaniem programu, który należało zoptymalizować, żeby działał możliwie szybko, zazgrzyta zębami oglądając zamieszczony wydruk, ale proszę o łagodny wymiar kary — program jest tak napisany, żeby łatwo było go przeanalizować i zrozumieć. Teraz zaproponuję zmiany, które przyspieszą jego wykonanie, chociaż przy okazji wydatnie zmniejszą jego czytelność. Należy zmienić lub dodać następujące linie



Rys. 4

```
15 LET A=COS(PI/4)
25 LET E=A*Y
27 LET C=(Y-70)*(Y-70)
34 LET D=X-70
40 LET Z=EXP(-1E-03*(C+D*D))
50 LET X1=X+E
60 LET Y1=Z+E
```



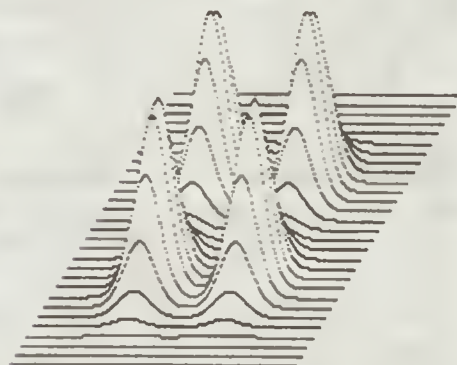
(linie 27 i 34 służą do przyspieszania obliczania wartości funkcji i nie są potrzebne, gdy w linii 40 znajdzie się inna, zaprojektowana przez użytkownika funkcja; skorzystaliśmy również z równości:

cos 45° = sin 45°

co oznacza, że jeśli ktoś chce zastosować inny kąt α musi postąpić trochę inaczej).

Teraz można już spokojnie uruchomić program. Działa on oczywiście długo — efekt końcowy będzie widoczny dopiero po kilkunastu minutach, ale trudno się spodziewać lepszego rezultatu — wartość wyrażenia w linii 40-tej jest liczona ponad 4000 razy!

Uważny czytelnik na pewno zauważył, że w porównaniu z rysunkiem nr 1 zrezygnowaliśmy z ramki, która otaczała deformowaną płaszczyznę. Nie ma to oczywiście żadnego znaczenia, a ułatwia napisanie czytelnego programu.

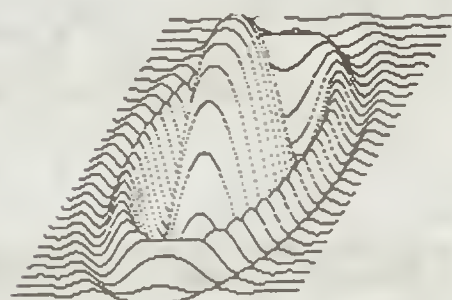


Jeśli ktoś ma wątpliwości, czy zasłanianie niewidocznych partii wykresu jest rzeczywiście potrzebne, niech zmieni w programie linię 70 na

70 PLOT X1, Y1

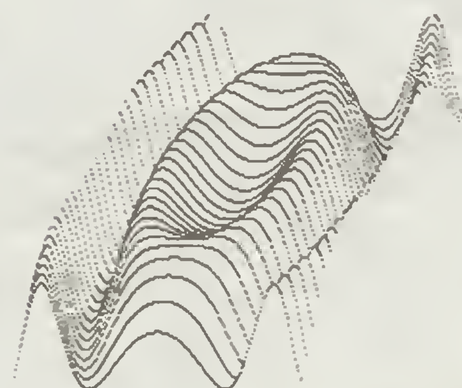
Teraz chyba nie ma wątpliwości, że warto zwiększać czytelność rysunku tak prostym chwytem!

Zaproponowany program może czasem dawać dziwne efekty — niektóre fragmenty powierzchni mogą być narysowane tak, jakby było widać ich spodnią stronę, zwłaszcza, gdy funkcja, której wykres rysujemy, rośnie dla zwiększających się wartości x. (Proponuję spróbować narysować wykres funkcji Z=70•EXP(−0.05 • (142−X)). Można dać sobie z tym radę przyjmując kąt α=90°, albo obracając wykres o 180° — np Z=70•(−0.05•X). Można zapewne zastosować również inne rozwiązania, ale to pozostawiam już inwencji Czytelników.



Uwagi dla użytkowników innych komputerów niż Spectrum:

a) jeżeli po uruchomieniu programu występuje błąd w linii 70 należy dopisać linię
   51 LET X1=INT(X1)
b) jeżeli wykres rysuje się na ekranie do góry nogami, należy zmienić instrukcję PLOT w linii 70 na
   PLOT 175−X1, Y1
c) jeżeli Twój komputer nie ma zdefiniowanej liczby π, należy dodać linię
   11 LET PI=3.14159
   lub zmienić w programie nazwę PI na 3.14159 we wszystkich miejscach gdzie ona występuje;
d) występujące w programie wartości 255, 141, 80 i 70 są dobrane dla Spectrum, toteż jeśli Twój komputer ma większą niż 176×256 punktów rozdzielczość, jego możliwości nie będą wykorzystane do końca.



Marcin Borkowski

Z COMMODORE NA MTP

Jak już informowaliśmy, podczas tegorocznych Międzynarodowych Targów Poznańskich swoje stoisko miał również Commodore. Gościł na nim oczywiście „Bajtek" ze swoim specjalnym wydaniem, które, jeszcze ciepłe, wprost z drukarni powieźliśmy do stolicy Wielkopolski. Jak widać na zdjęciu zainteresował się nim szczególnie Paul Moloney, dyrektor generalny Commodore Electronics Limited. A my mieliśmy okazję raz jeszcze, tym razem już bez ewidentnych błędów w sztuce fotograficznej, uwiecznić naszego interlokutora z 3 strony „Tylko o Commodore".

(go)

TWÓRCZE KOMPUTERY

Sztuka komputerowa oznaczała do niedawna poszarpane wizerunki, zaś muzyka komputerowa — kakofonię przypadkowych dźwięków. Dziś jedna twórczość wspomagana komputerem stała się już trendem dominującym. Muzycy zatrudnili do swej pracy komputery nie marnując ani jednego bita.

— Pięć lat temu znałem wszystkich którzy zajmowali się muzyką komputerową — twierdzi John Chowning, dyrektor amerykańskiego Centrum Badań Komputerowych, Muzyki i Akustyki. **Dziś mogą to robić wszyscy. Muzyka komputerowa wyszła na ulicę.**

Chowning sam jest częściowo odpowiedzialny za ten stan rzeczy: chociaż z wykształcenia kompozytor, jest autorem niektórych klubowych pojęć używanych w syntetyzatorach muzycznych.

Obecnie za pomocą standardowej wtyczki tanie syntetyzatory można podłączyć do każdego PC i już komputer zaczyna grać — jakość zależy od zastosowanego programu.

Tak więc nawet muzycy, którzy nie mieli do czynienia z komputerem mogą komponować, wykonywać i zapisywać utwory z pomocą tego urządzenia. Na przykład kalifornijski muzyk Dawid Grisman jest znany ze swoich kompozycji jazzowych na mandolinę. Jednak i on używa do komponowania komputera podłączonego do małego syntetyzera. Pisząc na swoim komputerze Grisman mógł przesłuchać wszystkie partie komponowanego kwartetu. Jednocześnie, zamiast wyobrażać sobie jak one brzmią razem. Efekt jest niezły: pianino lub flety, które udaje syntetyzator brzmią jak „prawdziwe" instrumenty muzyczne.

Następny krok do przodu w tej dziedzinie jest jeszcze większy: komputery zamiast muzyków! W programie komputerowym z Broderbund, który zdobył już wiele nagród, mali animowani muzycy grają na ekranie jazz, rock i utwory klasyczne. Operator gra razem z nimi uderzając w klawisze komputera i program idealnie integruje się z muzyką animowanych postaci. Prawdziwe jam session solisty z zespołu muzycznym.

Programista i trębacz Roger Dannenberg z Uniwersytetu Carnegie Mellon, wynalazł komputerowego akompaniatora dla poważnych muzyków. Program Dannenberga gra na przykład część orkiestrową „Błękitnej Rapsodii" na syntetyzatorze, podczas gdy solista — człowiek swoją partię na „normalnym" fortepianie. Komputer „słucha" muzyka jak prawdziwy akompaniator — to zwalniając, to przyspieszając — by utrzymać tempo, radzi sobie nawet wtedy gdy pianista opuści takt lub dwa.

na podst. Newsweek opr.

(pen)

GORĄCE, NIE DOTYKAĆ

Tak brzmiał po polsku ostrzegawczy napis umieszczony na głowicy każdej drukarki mozaikowej. Naukowcy, informatycy, matematycy i inżynierowie stoczyli prawdziwą batalię o obniżenie temperatury i poziomu emitowanego hałasu, a jednak nieostrożne obchodzenie się z drukarką ciągle grozi poparzeniem palców.

Większość głowic drukarek mozaikowych ma bardzo niską sprawność, która wynosi od 1 do 6 proc. Tarcie, promieniowanie elektromagnetyczne, prądy wirowe, rezystancja, zjawiska mechaniczne i akustyczne powodują, że 94 do 96 proc. dostarczanej energii zamienia się w ciepło, powodując nagrzewanie głowicy podczas pracy oraz obniżając jej żywotność.

Specjaliści pracujący nad problemem wyeliminowania tych niekorzystnych zjawisk sięgnęli po najnowocześniejszą metodę pracy twórczej: komputerowe wspomaganie projektowania CAD. Pomogło ono w udoskonaleniu głowicy drukarki i przystosowaniu do ogromnej szybkości druku, która wynosi 200 do 300 znaków na sekundę.

Zastosowanie komputera pozwoliło wszechstronnie przeanalizować prototypowy model głowicy i zoptymalizować go przy użyciu skomplikowanych metod matematycznych. Inżynierowie z amerykańskiej firmy Newbury Data Recording posłużyli się programem ANSYS oraz specjalnie opracowanym programem Sheba, co umożliwiło przejrzyste, graficzne opisanie zachowania się wszystkich ważnych elementów głowicy. Specjalistyczne programy CAD zaprzęgnięto do opracowania technologii produkcji na podstawie prototypowego modelu. W ten sam sposób przygotowano również rysunki detali i złożeniowe, zestawienia materiałów oraz skomplikowane trójwymiarowe przekroje pozwalające ocenić technologiczność konstrukcji.

Nowa głowica wykorzystuje efektywnie już 7,4 proc. dostarczanej energii. Wiedza zgromadzona przy jej opracowywaniu umożliwi w przyszłości dalsze polepszenie parametrów. Naukowcy zapowiadają zwiększenie szybkości druku, obniżenie temperatury oraz przedłużenie trwałości całego zespołu.

(j.j.)

KONKURS MATEMATYCZNY

ZADANIE NUMER 1

Trójkątem pitagorejskim nazywamy trójkąt prostokątny o długości wszystkich boków wyrażonej liczbą całkowitą np. 3:4:5 lub 5:12:13, itd.
W matematyce istnieje cały dział, oparty na tym pomyśle. Dzisiaj proponuję ułożenie programu, który w ciągu dwóch minut znajdzie większą liczbę takich trójkątów. Oczywiście NIE należy brać od uwagę trójkątów o bezpośrednim stałym mnożniku, których boki są proporcjonalne do boków mniejszego trójkąta
np. 6:8:10 lub 15:36:39
Do rozwiązania należy dołączyć program wraz z opisem oraz podać liczbę i otrzymane trójkąty pitagorejskie.
Program musi być napisany w języku BASIC. Dopuszczone są następujące instrukcje i funkcje:
PRINT, GOTO, GOSUB, RETURN, IF.. THEN, LET, DIM, FOR...NEXT...STEP, AND, OR, NOT, RND, ABS, INT, SQR, EXP, ATN, COS, SIN, TAN, LOG (logarytm naturalny), PI (liczba pi), SGN, +, −, *, /, ↑, >, >=, <, <=, =, <>.
UWAGA! Przy podstawianiu należy posługiwać się rozbudowaną instrukcją LET.
Ponadto od programu wymaga się:
1) program ma zaczynać się od linii 1 z przyrostem linii co 1
2) program zawiera po jednej instrukcji w linii.
3) obliczone boki trójkątów winny być przechowane w jawnie określonej tablicy
4) czas dwóch minut mierzy się od momentu wykonania pierwszego obliczenia (nie jest brany pod uwagę czas potrzebny na zadeklarowanie tablicy, dlatego Redakcja prosi o umieszczanie takich deklaracji na początku programu i zaznaczenie początku obliczeń przez np. PRINT"START --"
Jedyną więc instrukcją dopuszczoną przed instrukcją PRINT"START -->" jest DIM.
UWAGA!!! Wszystkie programy pretendujące do wygranej będą wpisane na jeden komputer, by jak zwykle zwyciężyła metoda rozwiązania problemu, a nie szybszy komputer. Zachęcamy więc do wzięcia udziału w konkursie i przysyłania swoich rozwiązań.

ZADANIE NUMER 2

Dziś pytanie będzie bardzo krótkie, choć pewnie przysporzy najwięcej kłopotów.
Pytanie brzmi następująco:
Z jaką dokładnością została policzona liczba Eulera, jeśli suma wszystkich cyfr po przecinku dała w rezultacie 4091 (cztery tysiące dziewięćdziesiąt jeden)???

Liczba Eulera definiowana jest jako granica ciągu (1 + (1/n))^n przy n dążącym do nieskończononości. Z dokładnością do 6 miejsc po przecinku liczba ta wynosi: 2.718281..... i jest używana między innymi jako podstawa logarytmów w komputerach. Tym którzy nie pamiętają wyżej poruszonych tematów (granica) proponuję sięgnąć do podręczników matematyki z ostatnich klas szkoły podstawowej i z pierwszych dla szkół średnich.
Do rozwiązania należy dołączyć program (obliczający liczbę E z dokładnością do n miejsc po przecinku? Wraz z opiem podać dokładność obliczenia liczba yw języku basic.

ZADANIE NUMER 3

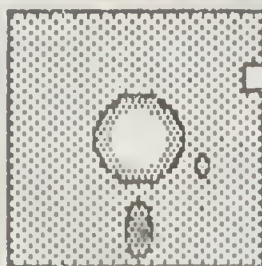
Proszę napisac program, który znajduje dwa takie ułamki proste, których iloraz daje w wyniku liczbę Π (Pi) obliczoną z jak największą dokładnością. Wystarczającą dokładnością niech będzie tym razem dokładność z jaką przechowywana jest liczba Pi w komputerach. Oczywiście wykluczone są ułamki, w których mianownik ma postać $m \cdot 10^n$ gdzie n jest liczbą naturalną a m jest liczbą (chodzi o liczby typu 100,3500,1200000, itp.)
Do rozwiązania należy dołączyć program wraz z opisem oraz podac otrzymany ułamek. Program musi być napisany w języku BASIC.

CENY NA DZIEŃ 25.06.88

	GIEŁDA „BAJTKA" (tys.zł)	PEWEX BALTONA (USD)	RFN (śred.) (DM)
SINCLAIR			
ZX-Spectrum 48 KB	120	115	90
ZX-Spectrum plus	155	—	99
TIMEX 2048	150	146	—
ZX Spectrum + 2	250	—	150
ZX Spectrum + 3	325	—	280
COMMODORE			
Commodore 64	230	219	299
Commodore 128	290	299	399
Magnetofon 1531	50	48	30
Stacja dyskietek 1541	240	—	355
Stacja dyskietek 1571	300	299	465
C-128 D	800	—	888
Drukarka MPS 801	190	—	—
AMIGA 500	1,2 mln	—	999
C-16	110	—	90
C-116	90	—	75
C-+4	170	—	150
ATARI			
ATARI 65 XE	170	125	100
ATARI 130 XE	210	199	230
Stacja dyskietek 1050	250	185	300
LDW 2000	300	199	—
XC-12	45	48	49
AMSTRAD			
464 mon. monitor	450	—	350
6128 z monit. monochromat.	510	—	680
64 kolor monitor	623	—	450
6128 mono. monitor	870	—	689
6128 kolor monitor	1,0 mln	—	988
Stacja dyskietek do 464	300	—	389
PC 1512 SD MD	1,1 mln	—	998
Dyskietki 3"	5	—	4-9
Dyskietki 5.25"	0.9	—	0.65
Dyskietki 3.5"	3-4	—	4-5.5

Sklep Bajtka ul. Koniewa 6 w BYTOMIU tel. 81-57-01

ZX-81	30.000
ZX-Spectrum 48 KB	110—120.000
ZX-Spectrum plus	150—160.000
Seikosha GP 50	100—110.000
Joystick	7—14.000
C64	260—270.000
C128	350—400.000
Magnetofon 1531	55—60.000
Stacja dyskietek 1541	250—260.000
Stacja dyskietek 1571	330—360.000
Drukarka	200—250.000
Dyskietki 5.25	1—1.600
Dyskietki 3.5	3—3.500
Atari 65 XE	170—180.000
Atari 130 XE	260—270.000
Atari 1050	260—270.000
ATARI 520 ST+SF 354	990.000



INDYWIDUALNY BANK DANYCH

Bartłomiej Szelerski, uczeń — 17 lat. Posiada komputer Atari 65 XE, magnetofon XC 12, stację dysków 1050. Oprogramowanie: około 100 gier oraz programy użytkowe. Proponuje wymianę doświadczeń, oprogramowania i literatury.
Adres: 62—600 Koło, ul. Wojciechowskiego 23/32

Władysław Kyryluk, lat 17. Posiada mikrokomputer bułgarski Pravetz-80, nawiąże kontakt w celu wymiany doświadczeń i oprogramowania. Korespondencja w języku polskim. Adres: ZSRR, Lwów-13, ul. Szczorska 8/9, 290013.

Tomasz Macki, posiada mikrokomputer Commodore 16. Proponuje wymianę gier i programów graficznych. Adres: 78—400 Szczecinek, ul. Sienkiewicza 2

Dariusz Zych, lat 15. Posiada ZX Spectrum 48 kB. Oprogramowanie: około 500 gier oraz programy użytkowe. Proponuje wymianę oprogramowania. Adres: 05—400 Otwock, ul. Szenwalda 2/4.

Tomasz Świątek, uczeń — 19 lat. Posiada mikrokomputer Schneider CPC 464. Oprogramowanie: programy do nauki matematyki. Proponuje wymianę oprogramowania oraz doświadczeń. Adres: 56—200 Góra Śl., ul. 1 Maja 48/14

Marek Lisowski, uczeń 16 lat. Posiada mikrokomputer Atari 65 XE, magnetofon XC 12 oraz stację dysków 1050. Oprogramowanie: gry, programy edukacyjne i graficzne. Proponuje wymianę oprogramowania na dyskietkach. Adres: 19—300 Ełk, ul. G. Zapolskiej 11

Wojciech Fiak, lat 12. Posiada Commodore C-64, stację dysków VC 1541 oraz organy. Oprogramowanie: programy muzyczne, matematyczne, graficzne oraz 160 gier. Adres: 42—600 Tarnowskie Góry, ul. Aleja Kwiatów 27/41

Mariusz Waściński, lat 12. Posiada Atari 800 XL, stację dysków 1050, pióro świetlne. Oprogramowanie: 120 gier oraz programy użytkowe. Adres: 37—450 Stalowa Wola, ul. 25-Lecia 6/52

Mariusz Juskowiak, lat 16. Posiada mikrokomputer Atari 520 ST wraz z monitorem. Oprogramowanie: gry, programy muzyczne, graficzne i narzędziowe. Pragnie nawiązać kontakt z użytkownikami komputerów serii ST w celu wymiany doświadczeń i oprogramowania. Adres: 78—400 Szczecinek, ul. Żukowa 5/3.

Grzegorz Morgiel, lat 17. Posiada ZX Spectrum +. Interesuje się informatyką oraz fantastyką. Proponuje wymianę oprogramowania, literatury oraz doświadczeń. Adres: 56—400 Oleśnica, ul. Klonowa 12/20

Krzysztof Dęga, uczeń szkoły średniej, lat 15. Posiada ZX Spectrum + z magnetofonem. Oprogramowanie: programy muzyczne, matematyczne oraz gry. Proponuje wymianę programów (posiada wiele nowości na Spectrum +). Adres: 64—920 Piła 4, ul. Bydgoska 61c/3

Waldemar Siwinski, student III roku Zarządzania. Interesuje się matematyką, informatyką oraz telekomunikacją. Posiada mikrokomputer Amstrad PC 1640 oraz drukarkę Amstrad DMP 4000. Nawiąże kontakt z użytkownikami tego typu komputera. Adres: 02—532 Warszawa, ul. [illegible]owa 10/3

Przedsiębiorstwo Zagraniczne KAREN
ul.Obrońców 23,
03-933 Warszawa
tel. 17 84 10
tlx 813948 kren pl

Szanowny Panie Dyrektorze,

Dziękujemy za zainteresowanie naszą firmą.
Z przyjemnością informujemy, że możemy zaspokoić wszystkie potrzeby Pana Przedsiębiorstwa określone w skierowanym do nas zapytaniu.
1. Oferujemy niezawodne i jednolite systemy komputerowe typu PC/XT/AT/386.
2. Instalujemy adaptery i oprogramowanie sieciowe ETHERNET.
3. Do Zakładu Poligrafii polecamy zestaw ATARI ST DESKTOP PUBLISHING - bogato oprogramowany i oczywiście z polskimi literami.
4. Do Klubu i Szkoły proponujemy ośmiobitowe ATARI XE.
Proszę nie niepokoić się o "wsad dewizowy" - to wszystko jest za złotówki.
Sprzęt objęty jest roczną gwarancją a przy odbiorze będzie mógł Pan uzupełnić swoje zbiory oprogramowania i literatury.

Z poważaniem,

DZIAŁ HANDLOWY

SB — 4

—ATARI—
ZX SPECTRUM

INSTRUKCJE
OPISY
LITERATURA

Szkoły i Kluby — Zniżka
Katalogi — Gratis
Co piąty program — Gratis
Wysyłka na cały kraj
Wypożyczalnia programów

D.H. „SEZAM" II p. g. 16.00—19.00
00-849 Warszawa UPT 66, skr.p. 14

D-1[illegible]

WOJEWÓDZKIE PRZEDSIĘBIORSTWO HANDLU WEWNĘTRZNEGO ODDZIAŁ W TYCHACH

VIDEOBIT

43-100 Tychy, aleja ZMP 77
tel. 27-69-75

- minikomputery 8-bitowe (Atari, Commodore, Schneider-Amstrad)
- minikomputery 16-bitowe kompatybilne z IBM PC
- drukarki 10" i 15" firm STAR, EPSON, AMSTRAD
- magnetowidy
- kamery video
- anteny satelitarne
- aparaturę badawczo-naukową

Zapewniamy o atrakcyjnych cenach.

[illegible]-7

UDOSKONALENIA

PROGRAMOWE I SPRZĘTOWE
DLA WSZYSTKICH MODELI **ATARI**
ORAZ KOMPUTERÓW PRACUJĄCYCH POD SYSTEMEM **MS-DOS**
WYSYŁA POCZTA
agencja mikro-komputerowa ***amico***
41-200 Sosnowiec P-157

[illegible]
NA ATARI XL/XE
[illegible] NA [illegible] U LUB ZA ZALICZ-NIEM POCZTOWYM
STUDIO — 8
[illegible]-300 MIELEC [illegible]
informacje + [illegible] + znaczek
[illegible]4

[illegible]
na ATARI i SPECTRUM
pisz do
[illegible] ATARBIT
skr. poczt. 7
[illegible] SOSNOWIEC 19
[illegible] 10 program GRATIS
+ kopertę + znaczek
otrzymasz katalog.
[illegible]
G — 94

Najnowsz[illegible] oprogramowanie ATARI, AMSTRAD, COMMODORE, S[illegible]UM oferuje „ATARES" — [illegible] z o.o. Ch[illegible], [illegible] 3 [illegible] 417-573 [illegible] 9.00—17.00 Sprzedaż wy[illegible]owa. Dla [illegible] i przedsiębiorstw oprogramowanie „pod [illegible]". Kupno — sprzedaż sprzętu kompu[illegible]owego i audiowizualnego — [illegible]. Dla instytucji rachu[illegible]

[illegible]-96

ASTRO — KOMPUTER STUDIO • PROGRAMY SHARP 700/800 — ATARI — SPECTRUM
Wrocław S.D.H. „ASTRA" w godz 15.00—18.00
oraz w sprzedaży wysyłkowej
54-515 Wrocław, Gdacjusza 39

G 82

SPECTRAVIDEO SVJ 328

Osoby posiadające programy użytkowe i gry proszone są o kontakt
Waldemar Tymoszuk
00-844 Warszawa
ul. Grzybowska 90 m 4

D-125

JAK REKLAMOWAĆ SIĘ W BAJTKU?

Reklamy przyjmuje Młodzieżowa Agencja Wydawnicza (Redakcja Wydawnictw Poradniczych i Reklamy), 04-028 Warszawa, Al. Stanów Zjednoczonych 53, pokój 313, tel. 10-56-82.
Cena reklamy biało-czarnej wynosi 300 zł za 1 cm². Do ceny podstawowej doliczane jest 30% za dodatkowy kolor i 100% w przypadku reklamy wielobarwnej. Ogłoszenie drobne kosztuje 200 zł za jedno słowo.

SPRZĘŻENIE ZWROTNE

Drogi Bajtku!

Na listy Czytelników odpowiada Marcin Waligórski

Swego czasu — wcale nie tak dawno — pisaliscie w „Bajtku" szeroko na temat produkcji sprzętu komputerowego w naszym kraju. Trzema wiodącymi tematami były: produkcja „Mazovii", drukarka D-100 oraz wprowadzenie do szkół masowo produkowanego komputera Elwro 800 „Junior". Z drobniejszych zapowiedzi wymienię tylko produkcję krajowych dyskietek i stacji dyskow. Po reportażach i komentarzach w optymistycznym tonie zapadła cisza. Polskich komputerow jak nie było, tak nie ma; teraz dodatkowo także nikt o nich nie mówi. Co się dzieje z „Mazovią", „Juniorem" i innym sprzętem komputerowym, ktory miał być w Polsce produkowany?

(nazwisko i adres do wiadomosci redakcji)

Losy trzech najgłośniejszych wyrobów naszego przemysłu mikroinformatycznego (o ile o takim można w ogóle mówić) przedstawiają się w telegraficznym skrócie następująco:

O spółce „Mikrokomputery" powołanej w celu produkcji Mazovii można dziś powiedzieć tyle, że góra urodziła mysz. Pomimo uczestnictwa w spółce największych potentatów naszego przemysłu elektronicznego i wieloletnich już zapowiedzi o rychłym rozpoczęciu produkcji, spółka zademonstrowała jak dotąd nieliczne egzemplarze Mazovii 1016 i 2016 przy okazji różnych wystaw i targów. Produkcja seryjna tych odpowiedników IMB PC/XT i AT idzie opornie, gdy na świecie standard przemysłowy ustalać poczynają komputery z procesorem 80386 — a więc o dwie generacje mlodsze. Właściwie można stwierdzić, że wobec silnej konkurencji dalekowschodnich „klonów" Mazovia ma teraz niewielkie szanse na zdobycie krajowego rynku.

Drukarka D-100 produkcji przedsiębiorstwa Mera-Błonie przez ostatnie dwa lata stała się konstrukcją przestarzałą pod każdym względem. Szansą na utrzymanie produkcji jest jednak ciągle zainteresowanie tym produktem krajów RWPG. Światowi potentaci w produkcji drukarek wytwarzają obecnie modele, które pod względem jakości, szybkości pracy i wyciszenia (oraz ciężaru!) biją D-100 na głowę.

Produkcja „Juniora" została rozpoczęta. Jest to na pewno przedsięwzięcie najbardziej z wymienionych zaawansowane i mające najlepsze widoki na przyszłość. Wszyscy są bowiem zgodni co do tego, że polski komputer szkolny musi być produkowany w kraju. Elwro boryka się jednak z problemami jakości oferowanych komputerów — zaś zamawiający którym jest Ministerstwo Edukacji Narodowej — nie kwapi się do płacenia za sprzęt nie dający gwarancji długowieczności. Spór trwa, a czas płynie szybko do przodu.

O krajowej produkcji dyskietek mówi się nadal w kategorii życzeń.

W związku z cyklem artykułów „Od środka" chciałbym przedstawić prosty sposob zabezpieczenia programu przed wczytaniem przez MERGE. W programie w BASIC-u, ktory ma być zabezpieczony, np.

```
10 LOAD ""CODE: LOAD ""CODE. RANDOMIZE USR 25000
```

wpisujemy przed pierwszym rozkazem cztery spacje. Następnie przy pomocy instrukcji POKE umieszczamy w komorkach 23755 do 23762 wartości 0, 0, 0, 0, 17, 8, 16, 8. Po wprowadzeniu tych wartości programu nie wolno już zmieniać, należy go od razu nagrac na tasmę.

Kejni

SPRZĘŻENIE ZWROTNE

Zastosowanej metody nie będę komentował, natomiast pozwolę sobie tylko na uwagę, że dobór adresu 23755 jest prawidłowy, o ile nie zmienialiśmy uprzednio wartości zmiennej systemowej PROG. W przeciwnym przypadku, np. gdy poniżej pamięci programu przechowujemy program w języku maszynowym, należy podane wartości umieścić w pamięci począwszy od komórek zawierających numer pierwszej linii programu.

Ponieważ jestem w programowaniu przedszkolakiem, więc mój program jest z natury rzeczy prymitywny. Niemniej umożliwia on rysowanie schematów elektronicznych. Program wyciągnąłem z szuflady wskutek napotkania w rubryce „Tylko dla przedszkolaków" materiałów poświęconych językowi Logo. Zdaje sobie sprawę z własnej niedoskonałości, niemniej program sprawdziłem w niezbyt zresztą skomplikowanych zastosowaniach.

Program uruchamia się procedurą SPIS

```
To SPIS
PR R — rezystor K — kondensator
PR T — tranzystor D — dioda
PR B — bezpiecznik M — masa
PR D — n — drut długość :n
PR N — nagraj schemat na taśmie
END

TO R
FD 3 LT 90
REPEAT 2 FD 2 RT 90 FD 10 RT 90
FD 2
RT 90 PU 10 PD FD 3
END

TO K
FD 3 LT 90 FD 3 BK 6 RT 90
PU FD 2 PD
LT 90 FD 6 BK 3 RT 90 FD 3
END

TO B
FD 3 LT 90
REPEAT 2 FD 2 RT 90 FD 10 RT 90
FD 2
RTt 90 FD 13
END

TO M
FD 3 RT 90 FD 5 LT 90
REPEAT 2 FD 5 RT 90 FD 1 RT 90
FD 5
RT 90 BK 5 LT 90 BK 3
END

TO D :n
FD n
END

TO T
FD 2 LT 90
REPEAT 36 FD 1 RT 10
RT 90 FD 4 LT 90
FD 3 BK FD 3 RT 90 FD 1
LT 45 FD 8 BK 8 RT 90 FD 8 LT 45
END

TO N
SAVESCR FIRST RL
END

TO O
LOADSCR FIRST RL
END
```

Marek Jaglarz
ul. Okólna 28/119
30-669 Kraków

Każdy, kto miał choć trochę do czynienia z językiem Logo dostrzeże od razu trywialność powyższego programu — z tego powodu skróciłem go, gdyż zamieszczenie wszystkich zamieszczonych w nim przez autora symboli przekracza możliwości „Sprzężenia zwrotnego". Dlaczego jednak list ów trafia na łamy tej rubryki a nie „Tylko dla przedszkolaków"? Otóż proszę zwrócić uwagę, że tylko program jest banalny — pomysł nie. Otóż wielokrotnie spotykałem się z pytaniami od osób uczących się języka Logo, do jakich praktycznych zastosowań może on służyć. Oto jedna z możliwych odpowiedzi na takie pytania, a jednocześnie ilustracja zasady, że najpierw trzeba mieć problem, a potem szukać sposobu jego lepszego rozwiązania przy pomocy komputera — a nie odwrotnie. Jest to również ilustracja faktu, że używać trzeba zawsze narzędzia w danej chwili najwygodniejszego. Nie ma sensu sięgać po OrCAD, jeżeli mamy tylko wykreślić schemat instalacji elektrycznej w mieszkaniu!

Bardzo dziękuję, za opublikowanie listów p. A. Szyszki (1,88) i p. R. Lipskiego (5/88) w sprawie CHUCKIE EGG. Pierwszy z nich zdopingował mnie, po stagnacji na 16 poziomie do osiągnięcia poziomu 24; drugi dał wiarę, że po poziomie 24 są następne.

Istotnie, po kilku probach i zastosowaniu odpowiedniej taktyki (śrubowanie wyniku punktowego na każdym poziomie) udało mi się stracić ostatnie, 49 życie na 33 poziomie (wersja Spectrum) z dorobkiem 440180 punktów! Zajęło mi to około 50 minut. I jeszcze kilka spostrzeżeń:

1) Za każde zebrane jajko przybywa 100 pkt, za kupkę ziarna — 50 pkt. Warto zbierać co tylko się da, bo w wersji Spectrum nie ma takiego ograniczenia, jak w Amstradzie i nie zdarza się przedwczesne zakończenie gry po zebraniu ostatniej kupki ziarna.

2) Dobrym treningiem jest granie przez jedną osobę za czterech graczy.

2) Dodatkowe życie uzyskujemy po uzbieraniu każdych kolejnych 10000 pkt. Pierwsze można uzyskać już po przejściu 3 poziomu bez utraty życia. Mój rekord na 1 poziomie — 2610 pkt, po 2 — 5980 pkt, po 3 — 10510 pkt, łącznie — z premiami.

4) Premia od 1000 pkt na 1 poziomie wzrasta do 9000 pkt. na poziomie 9 i potem już się nie zmienia.

5) Limit czasowy do 16 poziomu wynosi 900, od 17 do 32 — 800, a od 33 już tylko 700.

6) W wersji na Spectrum od 25 poziomu pojawiają się dodatkowe kury i lata kaczka. Od poziomu 33 wszystko jest tak, jak od 25, ale przy skróconym do 700 limicie czasowym; w szybszym tempie poruszają się kury i lata kaczka, zas tempo poruszania się farmera pozostaje bez zmian.

Przypuszczam, że tak jest do poziomu 40, co dalej — nie wiem.

Grażyna Błaszczyńska
ul. Tarczyńska 8 m 14
02-025 Warszawa

Zabawne, ile spostrzeżeń może przynieść jedna gra, i to w dodatku tak stara i prosta jak CHUCKIE EGG. Sam, co prawda nie doszedłem nigdy nawet do 10-poziomu, ale za to dla potencjalnych chętnych do zobaczenia „co jest dalej" mam dobrą radę. Zamiast wielokrotnie próbować dokonać tego przy pomocy joysticka, wystarczy jednokrotnie obejrzeć i zanalizować kod programu. Liczy on niewiele ponad 20 KB wraz z danymi, zatem pracy niezbyt dużo, zaś efekt gwarantowany — dowiedzieć się można dokładnie wszystkiego. A teraz kończę, bo muszę przecież w końcu dojść chociaż do tego 10-go poziomu.

Marcin Waligórski

---

2. Książki do „Klubu Książki — SOETO" zostały wybrane z planów wydawniczych na rok 1988. Uczestnicy Klubu będą je otrzymywać w miarę ich ukazywania się, czyli w ciągu całego 1988 roku (a jak podpowiada praktyka — również i w 1989 roku).

3. Książki będą wysyłane członkom Klubu za tzw. pobraniem pocztowym.

4. Należność za zamówione książki oraz koszt pobrania pocztowego płaci się listonoszowi przy odbiorze przesyłki z książkami, nie trzeba więc wnosić żadnych przedpłat z góry.

5. Obowiązkiem odbiorcy książek jest sprawdzenie, czy paczka jest nienaruszona. W wypadku wątpliwości należy odmówić przyjęcia. Po odbiór przesyłki należy wtedy zgłosić się do UPT, w którym otwarcia jej dokona pracownik pocztowy i ewentualnie sporządzi protokół. Tylko wtedy reklamacja może zostać uwzględniona.

6. Informacje dotyczące „Klubu Książki — SOETO" będą co pewien czas ukazywały się łamach „Bajtka".

Wyszczególnione na odwrocie książki są lub będą również do nabycia w n/w placówkach:

1. Centralna Składnica Harcerska Warszawa, ul. Mokotowska 24
2. Centralna Składnica Harcerska Szczecin, ul. M. Buczka 24c
3. Centralna Składnica Harcerska Bydgoszcz, ul. C. Skłodowskiej 26
4. P.P. „Dom Książki" (księgarnia techniczna) Wrocław, ul. Świdnicka 8
5. P.P. „Dom Książki" księgarnia Jelenia Góra, ul. 1-go Maja 16
6. P.P. „Dom Książki" księgarnia Wałbrzych, ul. Słowackiego 1
7. P.P. „Dom Książki" księgarnia Legnica, ul. Rynek 16/22
8. P.P. „Dom Książki" księgarnia Bydgoszcz, ul. Stary Rynek 15
9. Sklep firmowy „Bajtek" Bytom, ul. Koniewa 6
10. Biuro Handlowo-Techniczne „TARNOPOL" S.A. Tarnobrzeg, Pl. Bartosza Głowackiego 27

NUTKI

Cześć Maluchy!

Prawie każdy komputer może zamieniać się w instrument muzyczny. Jednym udaje się to lepiej, np. Commodore 64, inne zaledwie popiskują jak Spectrum. Muzykę programuje się na każdym mikrokomputerze inaczej, dlatego w naszej rubryce przeznaczonej dla użytkowników wszystkich mikrokomputerów znajdujecie wyłącznie programy bez dźwięku. Pomimo to zajmiemy się dzisiaj nauką muzyki.

Każdy z przedszkolaków poczułby się zapewne obrażony, gdybym zaczął mu tłumaczyć, że dźwięki zapisujemy za pomocą **nut**. Przypomnę tylko — tak na wszelki wypadek — że w zależności od czasu trwania dźwięku używamy nut o różnej wartości. Najdłuższy dźwięk, to **cała nuta**, która dzieli się na dwie **półnuty** lub cztery **ćwierćnuty**, osiem **ósemek**, szesnaście **szesnastek** albo też trzydzieści dwie **trzydziestodwójki**. Z kolei **półnuta**, to dwie **ćwierćnuty**, cztery **ósemki**, osiem **szesnastek** albo szesnaście **trzydziestodwójek**.

Proste i logiczne. Nie powinniście w takim razie mieć najmniejszych trudności ze zrozumieniem zasad naszej nowej gry. Dwóch zawodników stara się z nut o wartościach od trzydziestodwójki od półnuty utworzyć takt o łącznej długości całej nuty. Wygrywa ten, którego nuta zakończy takt. Nie można więc rozpoczynać gry od wybrania półnuty, bo wówczas przeciwnik wybierze również półnutę i zakończy grę.

Tym razem macie do dyspozycji dwa różne programy tej samej gry. Jeden napisany jest w **LOGO**, drugi w **BASIC-u**. Program w LOGO może być uruchomiony na każdym mikrokomputerze bez żadnych modyfikacji. Natomiast z programem w BASIC-u będą mieli kłopoty użytkownicy ATARI, gdyż ten komputer nie posiada tablic tekstowych. Amatorzy Commodore muszą jedynie zastąpić instrukcje CLS odpowiednimi komendami kasowania ekranu. Zwolennicy Spectrum i Amstrada mogą przepisać program bez żadnych zmian.

Na pierwszy rzut oka wydaje się, że strategia tej gry jest banalna. Spróbujcie sami, czy tak jest naprawdę. Przy okazji możecie porównać, który z programów jest bardziej przejrzysty, zwięzły i elegancki, który łatwiej modyfikować i rozwijać. Bardzo interesują mnie Wasze wnioski.

Romek

```
10 REM ################################
20 REM ############ Nutki ############
30 REM ################################

99 REM ###### dane początkowe ######
100 CLS
110 LET r=0:LET s=0
120 DIM n$(5,4)
130 DIM t(32)
140  FOR w=1 TO 4
150    FOR n=1 TO 5
160     READ n$(n,w)
170    NEXT n
180  NEXT w
190 DATA " ! "," - "," !"","  !"","  !""
200 DATA " ! "," ! "," ! ","  !"","  !""
210 DATA " ! "," ! "," ! "," ! ","  !""
220 DATA "O  ","#  ","#  ","#  ","#  "

299 REM ##### rysowanie planszy #####
300 CLS
310  FOR w=1 TO 4
320    FOR k=1 TO r
330     PRINT n$(t(k),w);
340    NEXT k
350   PRINT
360  NEXT w
370 PRINT
380 PRINT "Co wybierasz?"
390 PRINT " 1 - półnuta"
399 REM ###### ruch zawodnika ######
400 PRINT " 2 - ćwierćnuta"
410 PRINT " 3 - ósemka"
420 PRINT " 4 - szesnastka"
430 PRINT " 5 - trzydziestodwójka"
500 IF s>=1 THEN GOTO 700
510 INPUT n
520 LET r=r+1
530 IF n<>INT(n) OR n>5 OR n<1 THEN GOTO
 300
540 IF n=1 THEN LET s=s+1/2
550 IF n=2 THEN LET s=s+1/4
560 IF n=3 THEN LET s=s+1/8
570 IF n=4 THEN LET s=s+1/16
580 IF n=5 THEN LET s=s+1/32
585 IF n=6 THEN CALL &A000
590 LET t(r)=n
600 GOTO 300

699 REM ######## koniec gry ########
700 IF s>1 THEN PRINT "Przegrałeś, to wi
ecej niż cała nuta!":END
710 PRINT "Wygrałeś!"
```

r [illegible]
w [illegible]
n [illegible]
k [illegible]
o [illegible]
s [illegible]
t(32) [illegible]
n$(5,4) [illegible]

```
to kółko
 repeat 12 [lt 30 fd 5]
end

to kropka
 lt 45 fd 12 rt 45
 repeat 36 [lt 170 fd 20]
 rt 135 fd 12 lt 135
end

to ogonek
 rt 135 fd 6
 lt 35 fd 13
 rt 35 fd 6
 bk 6 lt 35
 bk 13 rt 35
 bk 6 lt 135
end

to półnuta
 kółko
 fd 60
 nowa.nuta
 make "suma :suma + 1 / 2
 test
end

to ćwierćnuta
 kropka
 fd 60
 nowa.nuta
 make "suma :suma + 1 / 4
 test
end

to ósemka
 kropka
 fd 60
 ogonek
 nowa.nuta
 make "suma :suma + 1 / 8
 test
end

to szesnastka
 kropka
 fd 50
 ogonek
 fd 10
 ogonek
 nowa.nuta
 make "suma :suma + 1 / 16
 test
end

to trzydziestodwójka
 kropka
 fd 40
 ogonek
 fd 10
 ogonek
 fd 10
 ogonek
 nowa.nuta
 make "suma :suma + 1 / 32
 test
end

to nowa.nuta
 pu
 bk 60
 rt 90
 fd 40
 lt 90
 pd
end

to test
 if :suma = 1 [wygrana]
 if :suma > 1 [przegrana]
end

to start
 cs ct
 pu
 lt 90 fd 300 rt 90
 pd
 make "suma 0
 pr [Wybierz i wpisz wartość nuty.]
end

to wygrana
 pr [Wygrałeś!]
end

to przegrana
 pr [Przegrałeś, to więcej niż cała nuta!]
end
```

LINEAR EXPRESSEM W XXI WIEK!

Na anglojęzycznym Zachodzie mówi się o nim The Bullet Train, czyli pociąg-pocisk. U nas rozpowszechniło się określenie Hikari. Japończycy zaś wszędzie tam, gdzie uwodzą cudami swej technologii umysły tłumów posługują się słowem shikansen.

Każda z tych nazw wywołuje jednak u milionów ludzi na całym świecie jednoznaczne skojarzenie-biały wąż wagonów o starannie zaprojektowanej aerodynamice, lokomotywa bardziej przypominająca nos pasażerskiego odrzutowca niż pojazd szynowy, a wszystko to przemieszczające się w niewiarygodnym tempie wśród typowego japońskiego krajobrazu. Shikansen nie jest już czymś nadzwyczajnym nawet dla nas, Polaków. Ekipa telewizyjna, towarzysząca Wojciechowi Jaruzelskiemu w jego podróży po Nipponie, pokazała każdemu Kowalskiemu jak mija czas we wnętrzu pociągu-pocisku. Uświadomiła tym, którzy codziennie dojeżdżają do pracy zdezelowanymi granatowo-żółtymi składami, że rozmowa telefoniczna z automatu zainstalowanego w Hikari jest czymś absolutnie normalnym.

Japończycy zdążyli się już do niego przyzwyczaić, a nawet znudzić. Nic w tym dziwnego. To już przecież 24 lata shikansen wozi ich na trasie Tokio-Osaka. Od momentu kiedy wyeksploatowano w przestrzeń ograniczoną torami pierwszy pocisk żelaznych szlaków, w 1964 roku, cała japońska super-szybka sieć kolejowa przewiozła prawie 100 mln ludzi. Mówi się i pisze na Wyspach Japońskich, iż Hikari powoli się starzeje. Technologie transportowe z początku lat sześćdziesiątych nie mogą być dzisiaj uznane za nowoczesne i wyatarczające (co na to PKP?). Coraz bardziej oblężone są shikanseny kursujące na newralgicznej dla gospodarki i handlu Nipponu trasie między stolicą a Osaką. Coraz częściej padają opinie, że Hikari jest za wolny w stosunku do fantastycznie podkręconej szybkości życia japońskich miast końca XX wieku. A przede wszystkim energochłonny, co dla zreprywatyzowanych japońskich kolei żelaznych ma dodatkowy wymiar.

Skoro są ludzie interesu gotowi płacić więcej za szybszą podróż, a mające swoje lata pociągi-pociski pękają w szwach, trzeba pomyśleć o ich następcach. Taka zapewne konstatacja stała się początkiem decyzji o praktycznych próbach systemów, jakie dotychczasowy symbol kolejnictwa — dwie stalowe szyny, mają okazję odesłać do muzeum techniki.

Nipponowi często zarzuca się swoistą rekordomanię. Ale przecież projekt pociągu, który na długie dziesięciolecia będzie znów najszybszym na świecie, powstał z konkretną myślą o zaspokojeniu rosnących potrzeb, a nie po to, aby wydzierać innym uzyskane w eksperymentalnych warunkach rekordy. Shikansen w 3 godziny pokonuje 300 mil dzielące Tokio od Osaki. 300 mil to około 480 km. Średnia prędkość Hikari na tym dystansie wynosi więc 100 mil na godzinę, a więc około 160km/h. Choć na prostej potrafi on dojść do 150 mil na godzinę, ok. 240km/h., to jednak przy użyciu konwencjonalnych materiałów technologii nawet niewiarygodnie zmodernizowanych, nie uda się z niego więcej wycisnąć. A przecież Japończykom śni się kolejowy wypad ze stolicy do Osaki, który nie zabrałby im więcej niż godzinę.

Palmę pierwszeństwa wydarł shikansenowi francuski TGV. 370km/ha to już spore osiągnięcie dla pojazdu wciąż niewolniczo przywiązane do zmienionych wprawdzie, ale wciąż obecnych stalowych prowadnic.

Jeszcze w latach sześćdziesiątych Japończycy postanowili zerwać ostatecznie z szynami i w walce z przestrzenią oraz czasem postawić wagony przyszłości na... poduszce magnetycznej. Idee pioniera profesora Erica Laithwaite z londyńskiego Imperial College na Dalekim Wschodzie zdecydowano przetransportować na lokalny grunt. Sercem wszystkiego miał się stać tak zwany silnik linearny — urządzenie bazujące na zjawisku znanym nawet uczniom szkoły podstawowej — odpychaniu się jednoimiennych biegunów magnesu.

Pozornie wszystko wydawało się dziecinnie proste — wzmocnić ową siłę, która sprawiała, iż dwa magnesowe krążki odpowiednio odwrócone pokonywały grawitację. A następnie niby konia pociągowego zaprząc ją do niewidzialnych dyszli lokomotywy. Nikt oczywiście nie wyobrażał sobie fabrykacji tak gigantycznych stałych magnesów. Nad ową instalacją należało przecież panować w każdej chwili. Do akcji wkroczyły więc odpowiedniej konstrukcji elektromagnesy. I wówczas Japończycy przekonali się, dlaczego świat nie rzucił się na nieskompilikowaną w teorii zasadę silnika linearnego.

Eksperymentalne urządzenia działały wedle powszechnie znanych reguł. Prąd płynął przez stosy cewek, a mimo to wytworzone siły nie były zdolne udźwignąć nawet połowy masy projektowanych wagonów i zapewnić im jakiejkolwiek rozsądnej szybkości. Powszechne ciążenie z łatwością zwyciężało elektromagnesy. Grawitacja wygrywała ze sztuczną lewitacją. Niepowodzenia na długie lata zatrzymały entuzjazm konstruktorów Nippon Railway. Nie zrezygnowali jednak i w ten sposób docignęli elektromagnetyczny pociąg do czasów zasadniczego przełomu w technologii produkcji materiałów stosowanych do przesyłania prądu elektrycznego.

Dopiero elektromagnesy wykonane ze zwojów nadprzewodowych przyniosły rozstrzygnięcie w latach osiemdziesiątych. Niewielki, prawdę mówiąc nieefektowny wagon ustawiony ni to na metalowej taśmie, ni to na czymś w rodzaju rampy pokazany na zewnątrz japońskiego pawilonu podczas wystawy EXPO-86 w kanadyjskim mieście Vancouver, nie wabił tłumów. Uważnego widza w osłupienie wprawić powinien jednak model całej tej maszynerii, ożywiający i działający zupełnie tak samo jak większy pierwowzór, na życzenie zwiedzającego. Miniaturowa koleika mknąca bez styku z podłożem po równie małej trasie — to była zapowiedź wielkiego sukcesu Japończyków.

Później były już próby z pasażerami — wagony unosiły się pod obciążeniem i przemieszczały trzymając „wysokość" 10 cm nad stalowym pasem. Techniczny korespondent brytyjskiego „Timesa" użył nawet określenia „niewidzialne gąsienice". I wreszcie nadeszła pora walki o szybkość. Prawdopodobnie już wkrótce dowiemy się, że supremacja Francuzów została przełamana.

Przy okazji przekonano się, iż nadprzewodnikowe elektromagnesy — nieporównanie silniejsze od dotychczas znanych, mają jeszcze jedną kolosalną zaletę — pobierają do pracy znacznie mniej energii. Znaczenie tego wniosku można będzie ocenić dopiero w przyszłości, bo gdy nastanie era nadprzewodników, metale superprzewodzące, jakie zastosowali Japończycy w swych linearnych silnikach umożliwią przesyłanie elektryczności na odległość ze znacznie mniejszymi stratami niż do tej pory. Miedź musi więc ustąpić ze swego pola.

Wróćmy wszakże do pociągu, jaki w Japonii zdążył już wejść do obiegowego języka Linear Express. Fachowcy z Japan Railway oszacowali ostatnio, iż dzięki nadprzewodnikom wydatki energetyczne na jednego przewiezionego z piorunującą szybkością docelową 300 mil na godzinę (ok. 480 km/h) pasażera będą niższe niż w przypadku nowoczesnych samolotów małego zasięgu, czy autobusów, lub samochodów (mniejsza masa pociągu nie jest też bez znaczenia). Takie kalkulacje są dodatkowym argumentem skłaniającym prywatnych inwestorów do przyspieszenia prób i zatwierdzenia planu ekspansji elektromagnetycznych, całkowicie wyciszonych linii na całe terytorium Nipponu. Stworzenie superszybkiej sieci połączeń będzie miało ogromne znaczenie dla kraju, którego pierwszoplanowym środkiem komunikacji jest nadal, na przekór wszystkim, właśnie pociąg. Wygląda na to, iż tak niegdyś romantyczny Kraj Wschodzącego Słońca wjedzie Linear Expressem w XXI wiek z prędkością zapierającą po prostu dech w piersiach. Wszystko bowiem wskazuje na to, że pierwsza linia magnetycznego łamacza rekordów ruszy już za dwa lata, a budowa systemu ogólnokrajowego ma być ukończona do 1997 roku.



Wojciech Łuczak